

Vysoká škola Báňská – Technická univerzita Ostrava

Fakulta strojní

Katedra automatizační techniky a řízení

Ovládání a vizualizace elektro- pneumatických prvků

Control and Visualization of Electro-pneumatic Components

Student:

Bc. Patrik Pilch

Vedoucí diplomové práce:

Ing. Miroslav Mahdal, Ph.D.

Ostrava 2017

Zadání diplomové práce

Student: **Bc. Patrik Pilch**
Studijní program: N2301 Strojní inženýrství
Studijní obor: 3902T004 Automatické řízení a inženýrská informatika
Téma: **Ovládání a vizualizace elektro-pneumatických prvků**
Control and Visualization of Electro-pneumatic Components
Jazyk vypracování: čeština

Zásady pro vypracování:

1. Popište výukový elektronický panel pro ovládání a měření stavů elektro-pneumatických prvků.
2. Ověřte funkčnost výukového panelu a všech jeho součástí, pomocí zapojení typových elektro-pneumatických úloh.
3. Seznamte se s vývojovou deskou Arduino Mega256, proveďte její připojení k elektronickému panelu a navrhnete algoritmus pro monitorování a ovládání připojených periférií k vývojové desce.
4. Navrhnete vizualizační aplikaci elektro-pneumatických úloh v prostředí SCADA/HMI, systému PROMOTIC 8.x, zjistíte možnosti komunikačního připojení k vývojové desce Arduino a navrhnete způsob předávání dat.
5. Zhodnoťte dosažené výsledky a možnosti jejich následného využití.

Seznam doporučené odborné literatury:

HEROUT, P. 2004. Učebnice jazyka C. 4.vyd. České Budějovice: KOPP a.s., 2004. ISBN: 978-80-7232-383-8.
KOPÁČEK, J. 1996. Pneumatické mechanismy. Díl 1. Pneumatické prvky a systémy. Skripta VŠB-TU Ostrava, 1996.
KOPÁČEK, J. 2005. Pneumatické mechanismy. Díl 2. Řízení pneumatických systémů. 2005. Skripta VŠB-TU Ostrava, 2005.
NEVRLÝ, J. 2003. Modelování pneumatických systémů. Akademické nakladatelství CERM, s.r.o., Brno, 2003. ISBN 80-7204-300-5.
PRZECZEK, S. 2014. Návrh a realizace elektro-pneumatických úloh. Bakalářská práce. Ostrava: VŠB – Technická univerzita Ostrava, Univerzitní studijní programy, 2014, 59 s.
ZMIJKA, M. 2013. Řízení pneumatického pohonu pomocí signálového procesoru. Diplomová práce. VŠB – Technická univerzita Ostrava, Katedra 352, 2013, 81 s.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Miroslav Mahdal, Ph.D.**

Datum zadání: 09.12.2016

Datum odevzdání: 15.05.2017



doc. Ing. Renata Wagnerová, Ph.D.
vedoucí katedry



doc. Ing. Ivo Hlavatý, Ph.D.
děkan fakulty

Místopřísežné prohlášení studenta

Prohlašuji, že jsem celou diplomovou práci včetně příloh vypracoval samostatně pod vedením vedoucího diplomové práce a uvedl jsem všechny použité podklady a literaturu.

V Ostravě:

.....

podpis studenta

Prohlašuji, že

- jsem byl seznámen s tím, že na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., autorský zákon, zejména § 35 – užití díla v rámci občanských a náboženských obřadů, v rámci školních představení a užití díla školního a § 60 – školní dílo.
- beru na vědomí, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen „VŠB-TUO“) má právo nevýdělečně ke své vnitřní potřebě bakalářskou práci užít (§ 35 odst. 3).
- souhlasím s tím, že diplomová práce bude v elektronické podobě uložena v Ústřední knihovně VŠB-TUO k nahlédnutí a jeden výtisk bude uložen u vedoucího diplomové práce. Souhlasím s tím, že údaje o kvalifikační práci budou zveřejněny v informačním systému VŠB-TUO.
- bylo sjednáno, že s VŠB-TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu § 12 odst. 4 autorského zákona.
- bylo sjednáno, že užít své dílo – diplomovou práci nebo poskytnout licenci k jejímu využití mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše).
- beru na vědomí, že odevzdáním své práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, bez ohledu na výsledek její obhajoby.

V Ostravě:

.....
podpis

Patrik Pilch
Topolová 213
73961 Třinec

Anotace

PILCH, P. *Ovládání a vizualizace elektro-pneumatických prvků: Diplomová práce.* Ostrava: VŠB - Technická univerzita Ostrava, Fakulta strojní, Katedra automatizační techniky a řízení, 2015, Vedoucí práce: Mahdal, M.

Semestrální projekt se zabývá popisem výukového panelu pro ovládání a měření stavů elektro-pneumatických prvků. V projektu je popsáno ověření funkčnosti jednotlivých prvků panelu a jeho následné opravy. Pro ovládání a monitorování elektronického panelu byla připojena mikroprocesorová jednotka a navrhnout její algoritmus. Dále je v projektu vyličen vizualizační software PROMOTIC, který pomocí ethernetu a komunikačního protokolu Modbus panel řídí. Závěrem byly vytvořeny čtyři vizualizační úlohy a jedna aplikace, ve které je možno sledovat a ovládat jednotlivé prvky panelu.

Klíčová slova: elektro-pneumatika, vizualizace, PROMOTIC, Arduino, Modbus

Annotation

PILCH, P. *Control and Visualization of Electro-pneumatic Components: Master thesis.* Ostrava: VŠB - Technical University of Ostrava, Faculty of Mechanical Engineering, Department of Control Systems and Instrumentation, 2017, Thesis head: Mahdal, M.

The master thesis deals with an instruction panel for controlling and measuring states of electro-pneumatics components. The project describes verification of the functionality of each panel element and its subsequent repairs. The microprocessor unit was connected for the controlling and monitoring of the electronic panel and then there was their algorithm created. Furthermore, the visualization software PROMOTIC is delineated in the project, which controls the panel by using Ethernet and the communication protocol Modbus. Finally, there were four visualization tasks and one application created, in which it is possible to track and control the individual elements of the panel.

Keywords: electro-pneumatic, visualization, PROMOTIC, Arduino, Modbus

Obsah

1	Úvod.....	8
2	Seznámení s výukovým elektronickým panelem.....	9
2.1	Tlačítka	10
2.2	Elektromagnetické relé	10
2.3	LED.....	12
2.4	Relé spínané časem.....	13
2.5	Digitální elektronický čítač.....	15
2.6	Mikroprocesorová jednotka	16
3	Ověření funkčnosti	19
4	Připojení desky Arduino k panelu.....	23
5	PROMOTIC	26
6	Propojení Arduino – PROMOTIC	29
7	Komunikace	31
7.1	Modbus	31
7.2	Datový model.....	33
7.3	Adresace	36
7.4	Nastavení komunikace v Arduino	37
7.5	Nastavení komunikace v softwaru PROMOTIC	38
8	Vizualizační aplikace	40
8.1	Diagnostický panel	40
8.2	Jednočinný pneumatický válec	41
8.3	Dvojčinný pneumatický válec	42
8.4	Vizualizace licího stroje	44
8.5	Vizualizace lisu.....	47
8.6	Vizualizace na mobilním zařízení se systémem Android.....	51
9	Závěr	52
10	Použitá literatura.....	55

Seznam použitých zkratk a symbolů

A – Proud [A]

AC – Alternating current – Střídavý proud

APU – Application data unit – Aplikační datová jednotka

COM – Communication port – Komunikační port

Datasheet – Manuál (dokumentace)

DC – Direct current – Stejnosměrný proud

EEPROM – Electrically Erasable Programmable Read-Only Memory – Elektronicky
mazatelná paměť

GND – Ground – Uzemnění

ICSP – In Circuit Serial Programming – Možnost programování kontroléru přímo v aplikaci
sériovou linkou

LED – Light-Emitting Diode – Dioda emitující světlo

MBAP - Modbus Application Protocol – Modbus aplikační protokol

MCU – Microcontroller unit – Mikroprocesorová jednotka

PDU – Protocol data unit – Protokolová datová jednotka

PLC – Programmable Logic Controller – Programovatelný logický automat

PWM – Pulse width modulation – Pulsně šířková modulace

RTU – Remote terminal unit – Řízené elektronické zařízení

RX – Receive – Příjem

SCADA – Supervisory control and data acquisition - Dispečerské řízení a sběr dat

TTL - Transistor-transistor logic – Tranzistorově-tranzistorová logika

TWI – Two Wire serial Interface – Dvou vodičové rozhraní

TX – Transmission – Vysílání

UART – Universal Asynchronous Receiver and Transmitter – Asynchronní sériové
rozhraní

USB – Universal serial bus – Univerzální sériová sběrnice

V – Napětí [V]

V_{in} – Voltage in – Vstupní napětí

W – Výkon [W]

Ω – Odpor [Ω]

1 Úvod

Vizualizace procesů a jejich vzdálené řízení je v dnešní době již standardem pro téměř jakékoliv průmyslové firmy. Tato oblast se neustále vyvíjí a její růst bude nepochybně i v následujících letech.

Diplomová práce se bude věnovat elektro-pneumatickému výukovému panelu, jehož elektrická část byla vytvořena v rámci bakalářské práce na Vysoké škole báňské – Technické univerzity Ostrava, katedře Automatizační techniky a řízení.

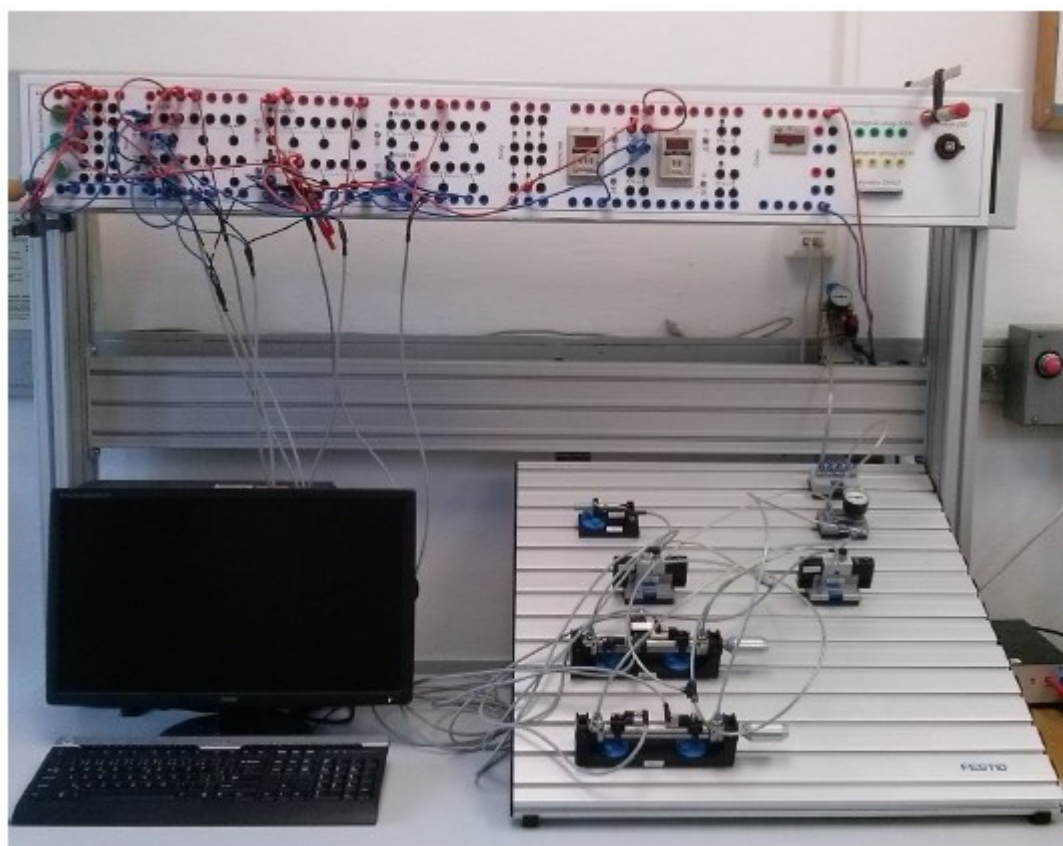
V rozsahu tohoto projektu budou vypsány prvky elektro panelu a jejich základní vlastnosti. Následně bude provedeno ověření, zdali je panel plně funkční a případné nedostatky opraveny.

Panel bude připojen k mikroprocesorové jednotce Arduino, díky které bude možno jednotlivé prvky panelu ovládat případně snímat jejich aktuální stav. Popsán bude také vizualizační software PROMOTIC, který slouží k řízení a zobrazování technologických procesů. Dále budou analyzovány možnosti připojení mikroprocesorové jednotky k počítači a s tím související typ komunikace se softwarem PROMOTIC. Vybrané připojení bude následně provedeno a naprogramováno.

Závěrem budou vytvořeny vizualizační aplikace v prostředí SCADA/HMI, systému PROMOTIC. Základní aplikací bude diagnostika panelu, dále budou vytvořeny jednoduché úlohy se základními elektropneumatickými prvky a ještě také již konkrétní simulační úlohy zobrazující určitý proces.

2 Seznámení s výukovým elektronickým panelem

Vytvořený panel je zabudován ve výukovém elektropneumatickém stole firmy Festo, který se v poslední době hojně využívá na vysokých a středních školách a dává tak možnost vyzkoušet si naučené teoretické znalosti v praxi. Jedná se o výukový program systémů a odborného vzdělávání. Důvod k vytvoření nového panelu a nepoužití originálních prvků od Festa, které tyto výukové stoly standardně doplňuje, byl rozšířit jeho další možnosti využití. V tomto případě jde o přidání mikroprocesorové jednotky, která bude umět získávat informace o jednotlivých prvcích panelu a případně je i ovládat. Jednotka se dále bude starat o přenos dat mezi panelem a počítačem (a naopak), kde se budou díky vizualizačnímu nástroji jednotlivé stavy a informace z panelu zobrazovat.

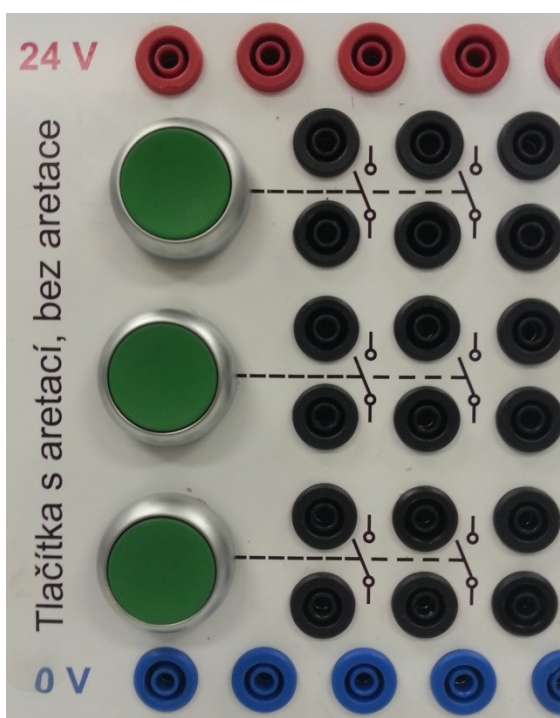


Obr. 1 Výukový stůl Festo Didactic s elektronickým panelem (Przeczek, 2016)

2.1 Tlačítka

Na panelu jsou k dispozici tři tlačítka. Dvě jsou s aretací (mají dva stabilní stavy, po stisknutí zůstanou v poloze zapnuto) a jedno tlačítko je bez aretace (monostabilní). Ovládací hlavice tlačítka má označení M22-DL-G, která je díky upevňovacímu adapteru M22-A přichycena k panelu spolu se spínacím kontaktním prvkem M22-K01. Výrobce těchto prvků je firma Eaton Moeller.

Spínací jednotka je při použití stejnosměrného napětí 24V schopna přenést proud až 3A.



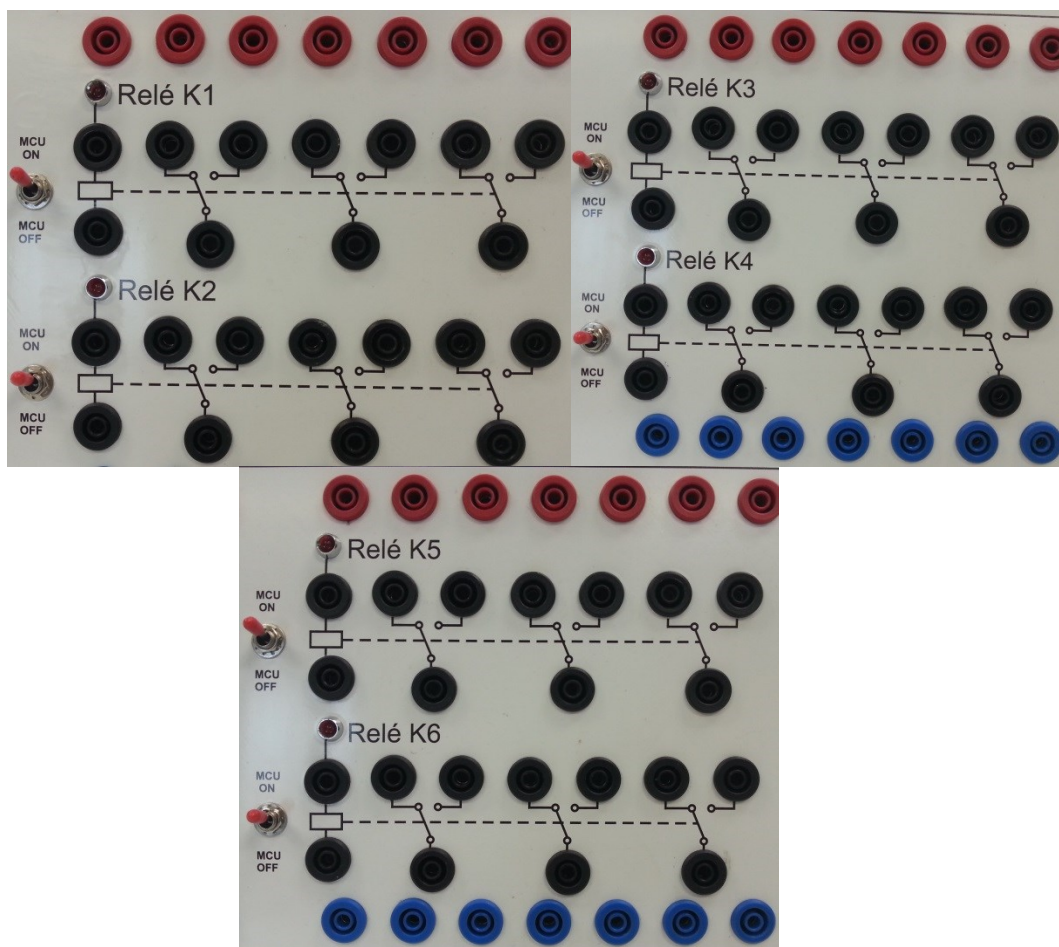
Obr. 2 Tlačítka

2.2 Elektromagnetické relé

K dispozici je šest spínacích elektromagnetických relé (dále jen relé), díky kterým můžeme nepřímě spínat různé elektrické obvody. Jedná se o elektromagnetické relé se stejnosměrnou cívkou. Po připojení zdroje na cívku, se vyvolá magnetický tok, který přitáhne kotvu a ta sepně pružné kontakty.

Každé relé ještě v obvodu obsahuje ovládací relé. To umožňuje spínat hlavní kontakty přes mikroprocesorovou jednotku. Jelikož výstup z procesoru je maximálně 5V, byla nutná tato úprava, která zajistí, že po sepnutí ovládacího relé pomocí 5V se na cívku hlavního

relé dostane 24V, které panel používá. K výběru, zda budeme hlavní relé spínat přímo na panelu, nebo nepřímo z procesoru slouží přepínač MCU ON/OFF.



Obr. 3 Relé

Jako hlavní relé je použito N4100CHS3DC24AC od výrobce Forward Relays, jehož parametry jsou uvedeny v tabulce níže.

Tab. 2.1 Vlastnosti hlavního relé

<u>N4100CHS3DC24AC</u>	
Vlastnosti cívky:	
U [V]	24
R [Ω]	2880
P [W]	0,2
Vlastnosti kontaktu	
Typ	1 x sepnutí
Materiál	Ag
I [A]	5A
Doba sepnutí [ms]	5
Doba rozepnutí [ms]	5
Mechanická životnost	Min 10^7 operací

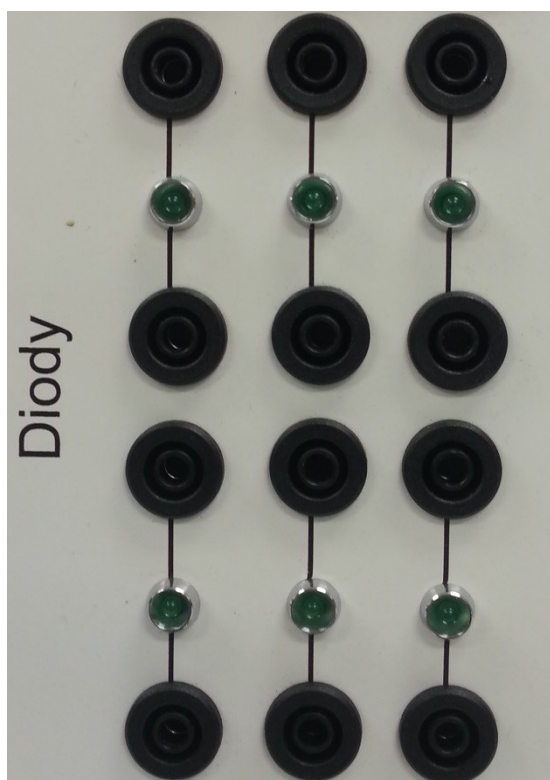
Ovládací relé je použito FTR-F3AA005E od výrobce Takamisawa.

Tab. 2.2 Vlastnosti ovládacího relé

<u>FTR-F3AA005E</u>	
Vlastnosti cívky:	
U [V]	5
R [Ω]	125
P [W]	0,2
Vlastnosti kontaktu	
Typ	1 x sepnutí
Materiál	Ag
I [A]	5A
Doba sepnutí [ms]	10
Doba rozepnutí [ms]	10
Mechanická životnost	Min 5×10^6 operací

2.3 LED

Každé hlavní relé má k sobě paralelně připojenou červenou LED diodu, která signalizuje, že jsou kontakty sepnuty. K dispozici je zde ještě šest kusů LED, které můžeme do obvodu připojit například z důvodu, abychom zjistili, jestli je v tomto místě napětí.



Obr. 4 Diody

LED jsou již uzpůsobeny k použití průmyslového napětí 24V a to díky sériovému připojení rezistoru.

Proud, který může diodou protékat je 7mA a napětí 2V. Bylo potřeba vybrat rezistor, který zajistí úbytek napětí 22V.

$$R = \frac{U_R}{I_D} = \frac{22}{0,007} = 3142,85 \quad (1)$$

Nejbližší rezistor, který je v řadě hodnot a dá běžně koupit je 3,3kΩ. Ten byl i použit v tomto zapojení.

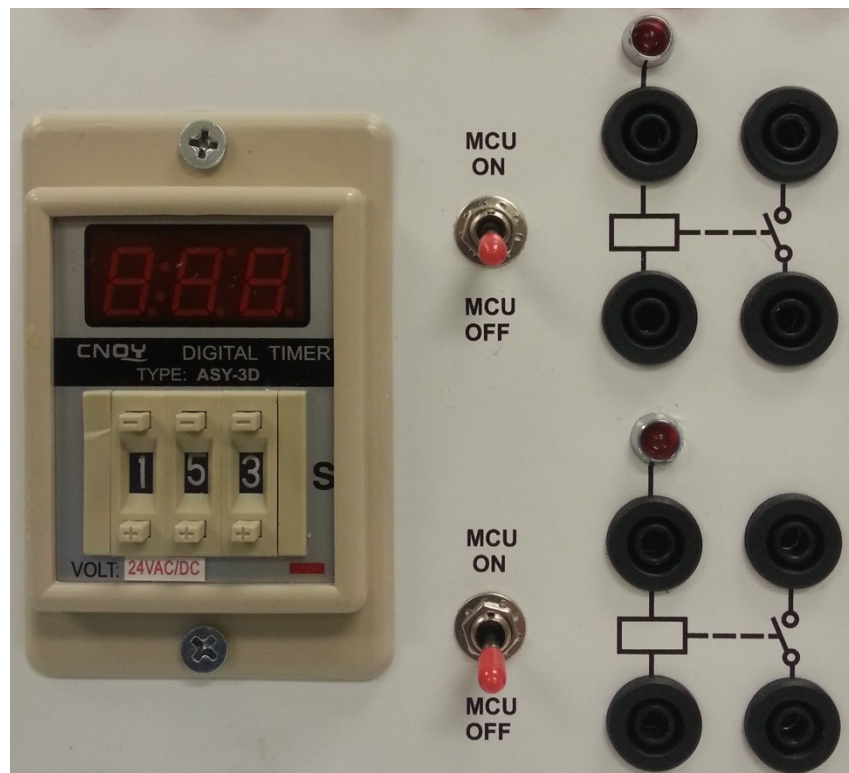
2.4 Relé spínané časem

Relé spínané časem umožňují spínat obvody až po tom, co uběhne čas, který byl nastaven. Čas se začne automaticky odpočítávat, pokud na relé převedeme vstupní napětí.

Elektronický panel obsahuje dvě jednotky časových relé, které se liší v časových rozsazích, které jdou manuálně zadat tlačítky pod sedmi segmentovým displejem. Displej zobrazuje uplynulý čas od zapnutí jednotky. Jestli se stopky ještě nedostaly k nastavenému času, poznáme podle blikající identifikační diody. Po sepnutí výstupní části dioda blikat

přestane. I zde je možnost nastavit podle vypínače MCU ON/OFF, zda bude relé spuštěné manuálně na panelu nebo mikroprocesorovou jednotkou. Vstupní napětí je 24V.

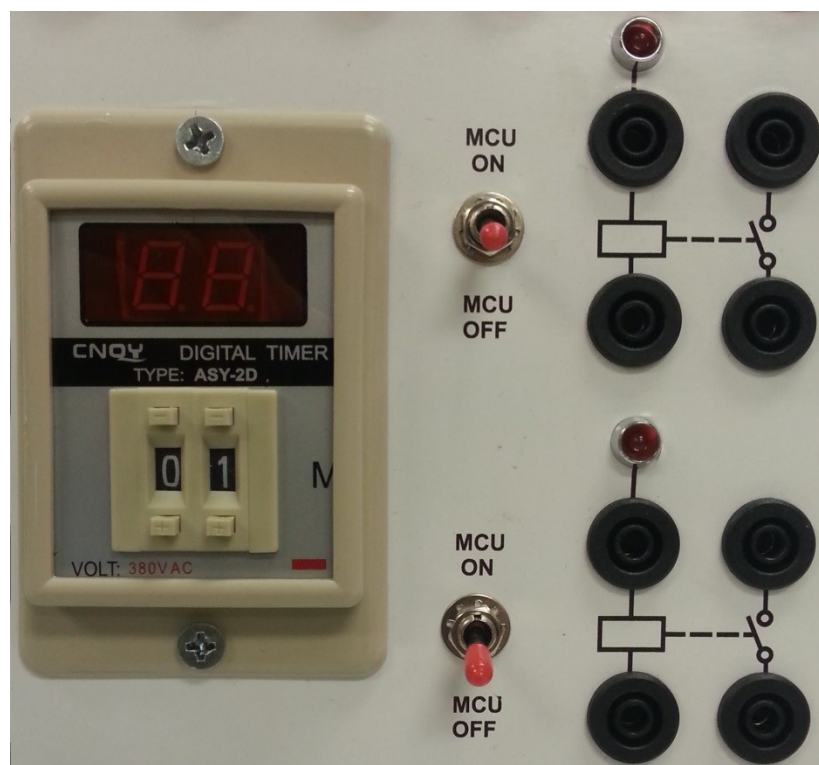
První relé má označení ASY-3D a jeho časový rozsah je 0-99,9 sekund.



Obr. 5 Sekundové časové relé

Po uplynutí času se sepnou dvě relé. Můžeme tedy spínat dva odlišné obvody.

Druhé časové relé má modelové označení ASY-2D a jeho rozsah je 0-99 minut. Ostatní parametry jsou stejné, jako relé sekundové.



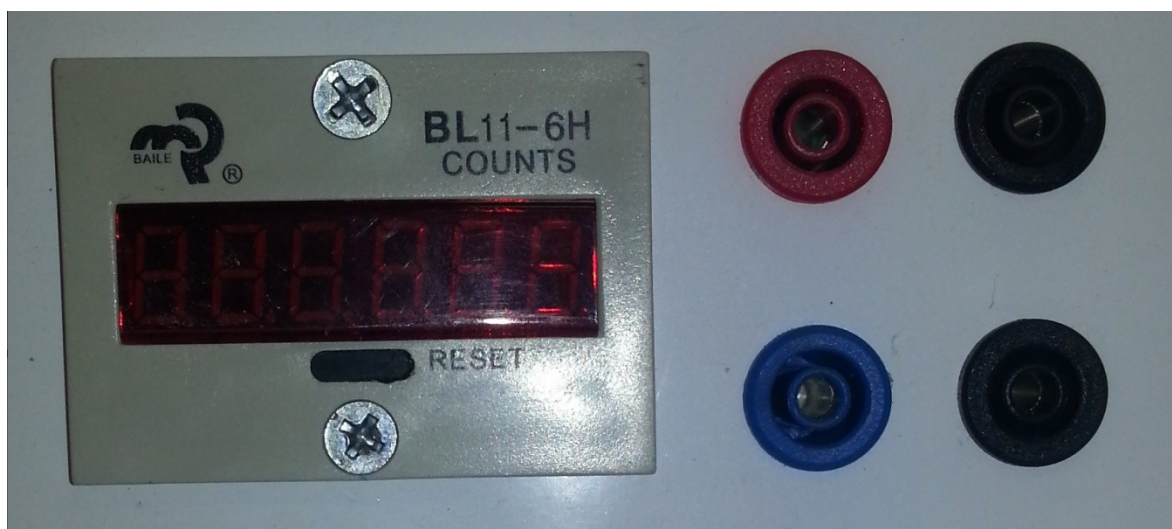
Obr. 6 Minutové časové relé

Tab. 2.3 Vlastností časového relé

Napájecí napětí	DC 12V-48V AC 24V-380V 50/60Hz
Odolnost výstupních kontaktů	5A (AC 220V)
Operační teplota	Od -10°C do 50°C
Vlhkost	Od 35% do 85%
Rozměry	57,7 x 82 x 87 mm
Váha	≈150g

2.5 Digitální elektronický čítač

Čítač je přístroj, díky kterému můžeme počítat, kolikrát proběhl určitý proces nebo událost. Konkrétně čítač umístěný na panelu má sedmi segmentový displej, který umí zobrazit až 999999 proběhnutých pulzů. Jeho modelové označení je BL11-6H a výrobce tohoto přístroje je Baile. Provozní napětí 24V. K dispozici je na čelní straně resetovací tlačítko, které čítač vynuluje.



Obr. 7 Čítač

Tab. 2.1 Vlastnosti čítače

Napájecí napětí	DC 24V
	AC 220V
Operační teplota	Od -10°C do 40°C
Rychlost odezvy	50ms
Rozměry	56 x 40 x 90 mm
Váha	≈65g

2.6 Mikroprocesorová jednotka

O komunikaci s počítačem se zde stará stavebnice Arduino MEGA 2560. Jde o Open-Source platformu, která je založena na jednoduchém hardwaru a softwaru a používá se pro snadný návrh a vývoj elektronických programovatelných zařízení. Velkou výhodou Arduino platformy je jednoduchá aplikace, velmi velký zástup kompatibilního hardwaru a značná podpora komunity. Oproti ostatním stavebnicím se Arduino vyjímá také velice příznivou cenou.

Srdcem této desky je procesor ATmega2560 vyrobený firmou Atmel. Tento 8mi bitový procesor disponuje 256kB programovatelnou flash pamětí (použitá pro uložení kódu), 4kB EEPROM a 8kB SRAM (Arduino MEGA 2560, 2015).

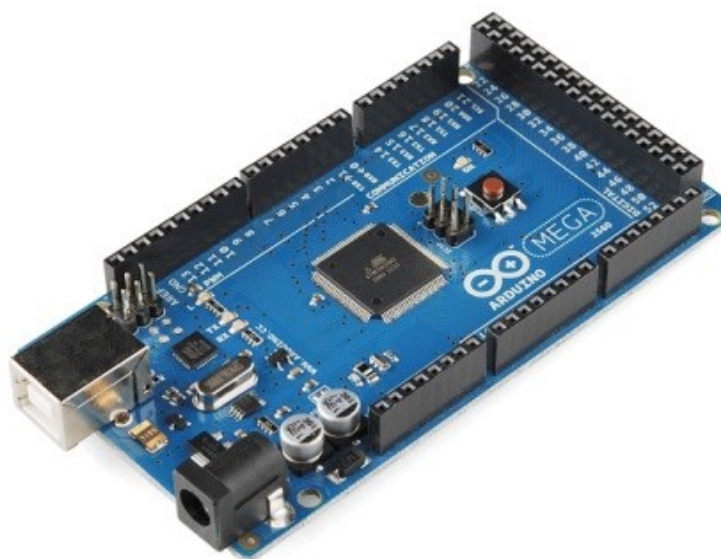
Na desce dále najdeme 16MHz krystalový oscilátor, 54 digitálních vstupně/výstupních pinů (z toho 15 pinů může být použito jako PWM výstupy), 16 analogových vstupů, 4x

UART (hardwarové sériové porty), USB konektor, napájecí jack, ICSP (programování obvodu bez nutnosti jeho vyjmutí a vložení do speciálního zařízení, které k tomuto účelu slouží) a resetovací tlačítko.

Provozní napětí je 5V, doporučené vstupní napětí 7-12 V s tím, že limitní napětí je 6 a 20 V. Vstupně/výstupní piny lze zatížit 20 mA a 3,3 V pin lze zatížit 50 mA. Deska plošných spojů je 101,52 mm široká, 53,3 mm vysoká a její váha je 37 g.

Megu 2560 můžeme programovat pomocí Arduino Software (IDE), které je volně ke stažení na webu výrobce. Na stránkách jsou k dispozici i objemné tutoriály, část z nich je implementována v softwaru. Programuje se jazykem odvozeného z jazyka Wiring. Další možností jak do procesoru nahrát program, je použít bootloader, který nám umožní nahrát nový firmware externím programátorem.

Stavebnice může být napájena USB konektorem nebo externím napěťovým zdrojem. Výběr zdroje napájení je proveden automaticky. Jako externí zdroj se používá AC/DC adaptér nebo baterie. K připojení slouží 2,1 mm konektor. Pokud je Arduino napájeno napětím menším než 7 V, 5 V piny mohou mít napětí menší než pět voltů a deska může být nestabilní. V případě použití napětí 12 V a více, může dojít k přehřátí regulátoru a následně k jeho poškození.



Obr. 8 Arduino MEGA 2560 (POLOLU, 2016)

Každý z 54 digitálních pinů může být jako výstup nebo vstup. Záleží na použití funkcí *pinMode()*, *digitalWrite()* a *digitalRead()*. Pracují s napětím 5 V a každý pin dokáže dodat nebo přijmout proud 20mA.

Některé piny mají speciální funkce:

- Sériová linka
 - 0 – pin 0 (RX) a pin 1 (TX)
 - 1 – pin 19 (RX) a pin 8 (TX)
 - 2 – pin 17 (RX) a pin 16 (TX)
 - 3 – pin 15 (RX) a pin 14 (TX)
- Externí přerušení
 - Přerušení 0 – pin 2
 - Přerušení 1 – pin 3
 - Přerušení 2 – pin 21
 - Přerušení 3 – pin 20
 - Přerušení 4 – pin 19
 - Přerušení 5 – pin 18
- SPI komunikace
 - MISO – pin 50
 - MOSI – pin 51
 - SCK – pin 52
 - SS – pin 53
- LED
 - Zabudovaná na desce, spojena s digitálním pinem 13
- TWI
 - SDA – pin 20
 - SCL – pin 21

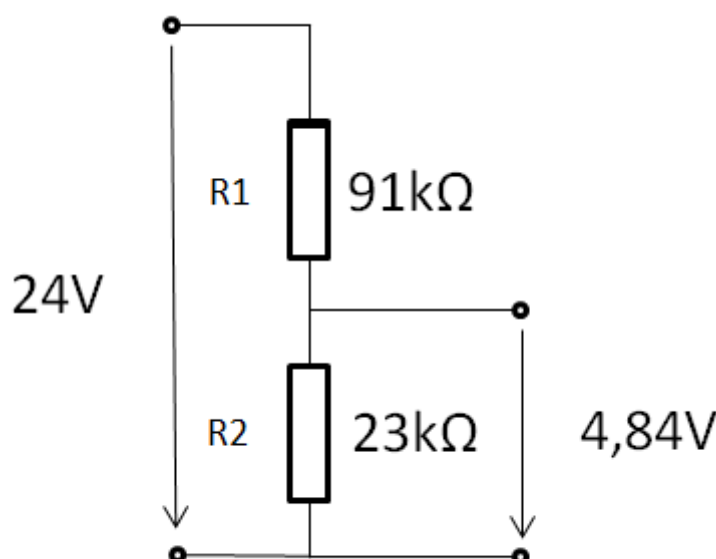
Dále má MEGA 2560 16 analogových vstupů, kde každý z nich má rozlišení 10 bitů (1024 rozdílných hodnot, rozlišení 0,0049 V). Standardně měří ze země na 5 V, ale pomocí pinu AREF a funkce *analogReference()* lze rozsah změnit na sestupnou hranu.

3 Ověření funkčnosti

Jelikož byl panel vytvořen v rámci bakalářské práce a pro jeho další rozšíření je potřebná plná funkčnost, provedla se kontrola všech komponent jednotlivě. Bohužel už po pár minutách testování bylo jisté, že ne vše je zcela v pořádku a bude potřeba tomu věnovat více času, než se předpokládalo.

Jako první byly otestovány tlačítka, o kterých můžeme konstatovat, že jsou v pořádku. Následně došlo k testování 6 kusů relé. Při tomto zkoušení nastal první problém. Z panelu se začal linout zápach, který signalizoval, že se něco pálí. Bylo nutné tedy panel rozebrat.

Po rozebrání panelu bylo ihned pouhým okem vidět, kde je problém. Jednalo se o napěťové děliče, které snižují napětí z 24 V na 5 V u časových relé.



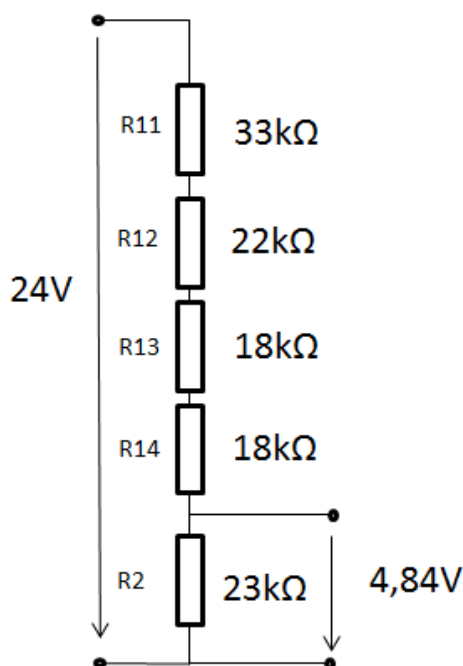
Obr. 9 Napěťový dělič

Naprosto spálený byl odpor R_1 . Prvotní úvaha byla, že na tomto odporu dochází k příliš velkému úbytku napětí (cca 19 V), který tento typ rezistoru nevydrží. Provedla se tedy náhrada jednoho rezistoru na rezistory čtyři, které byly řazeny sériově. Jejich hodnoty byly zvoleny tak, aby celkový součet byl opět 91 kΩ: $2 \times 18 \text{ k}\Omega + 33 \text{ k}\Omega + 22 \text{ k}\Omega$.

$$U_2 = U_1 \cdot \frac{R_1}{R_2 + R_1} = 24 \cdot \frac{23 \cdot 10^3}{23 \cdot 10^3 + 91 \cdot 10^3} = 4,84 \text{ V} \quad (2)$$

Po opětovném zapojení děliče se již nic nepálilo, nicméně obvod nebyl stále v pořádku. V místě, kde má být napětí cca 4,84 V byla naměřena nula. Nakonec bylo

zjištěno, že byl chybně nainstalován rezistor R_2 na desce plošných spojů. Místo rezistoru s hodnotou $23\text{ k}\Omega$ byl zde připájen rezistor s odporem pouhých $23\text{ }\Omega$. Po prozkoumání byla tato chyba na všech čtyřech napěťových děličích. Po výměně za správné rezistory (s hodnotou $23\text{ k}\Omega$) již bylo vše v pořádku a na výstupu děliče se objevilo napětí, které bylo vypočítáno.

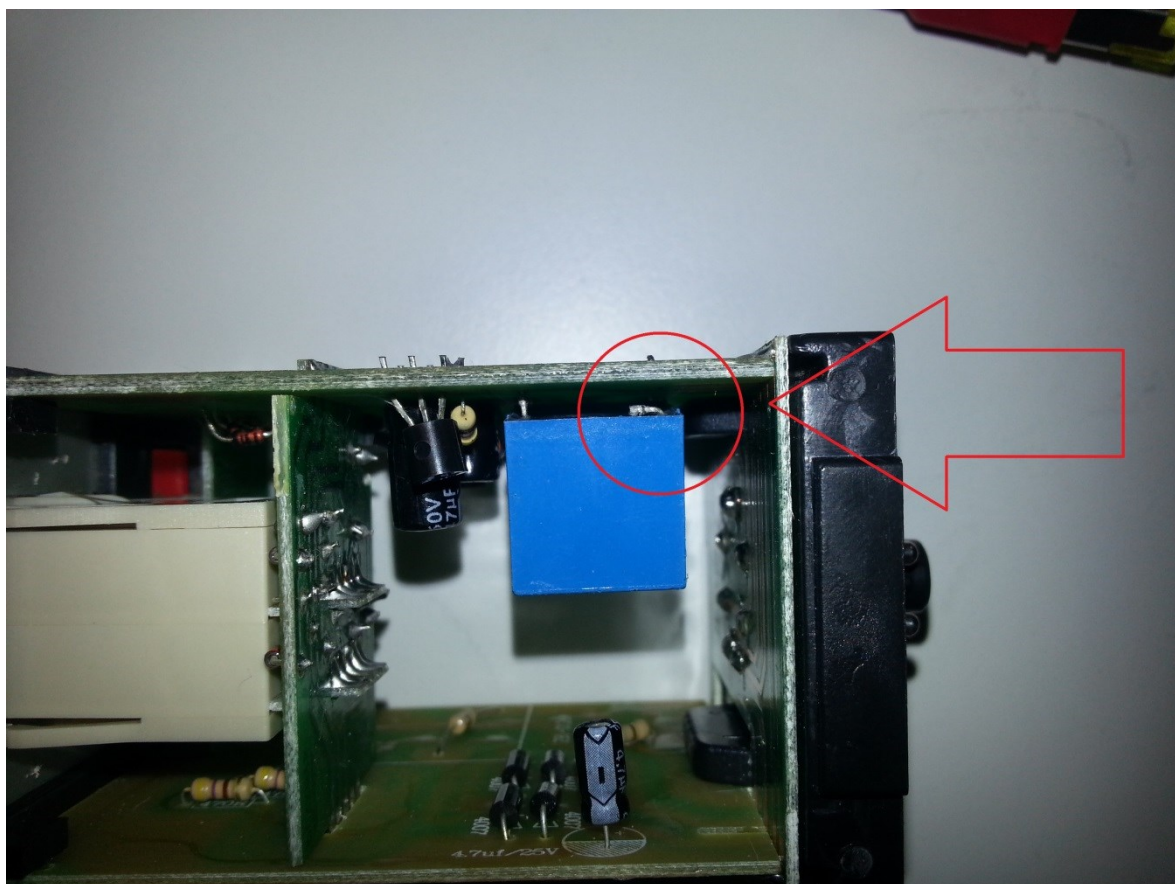


Obr. 10 Opravený napěťový dělič

Další problém, který se u panelu objevil, byl ten, že u minutového časového relé se po uplynutí času nesepnul jeden ze dvou oddělených výstupů. Po zjištění, že veškerá kabeláž mezi konektory, panelem a časovým relé byla v pořádku, zbývalo už jen, že problém je v samotné jednotce časového relé a bude nutné koupit relé nové.

Jelikož ale v České republice nebylo k sehnání a ze zahraničí by doručení trvalo dlouhou dobu, zkusilo se relé rozebrat. Díky důkladnému měření se zjistilo, že je vadné pouze jedno z elektromagnetických relé na desce plošných spojů.

Součástky do těchto desek jsou většinou ukládány a pájeny strojově. Při sundávání tohoto vadného relé se zjistilo, že celá chyba je pouze v jednom z pinů. Ten při usazování na desku nebyl pravděpodobně zcela rovný a neprošel připraveným otvorem. Neměl tedy potřebný kontakt s vodivými cestami.



Obr. 11 Chybně nainstalované relé

Další problém se objevil u desky plošných spojů od šestice kusů relé. Jelikož ne všechny spínaly výstupy a po kontrole kabeláže, která byla v pořádku, se pozornost zaměřila na funkčnost desky. Po proměření, se zjistilo, že cesty, které by měly být vodivé, obsahují velmi malou vrstvu mědi a tím pádem byly některé přerušené. Takových míst se objevilo několik. Opravilo se to přidáním trochu cínu mezi přerušovaný kontakt. Chyba pravděpodobně vznikla při špatném osvětlení fotocitlivé vrstvy desky plošného spoje.



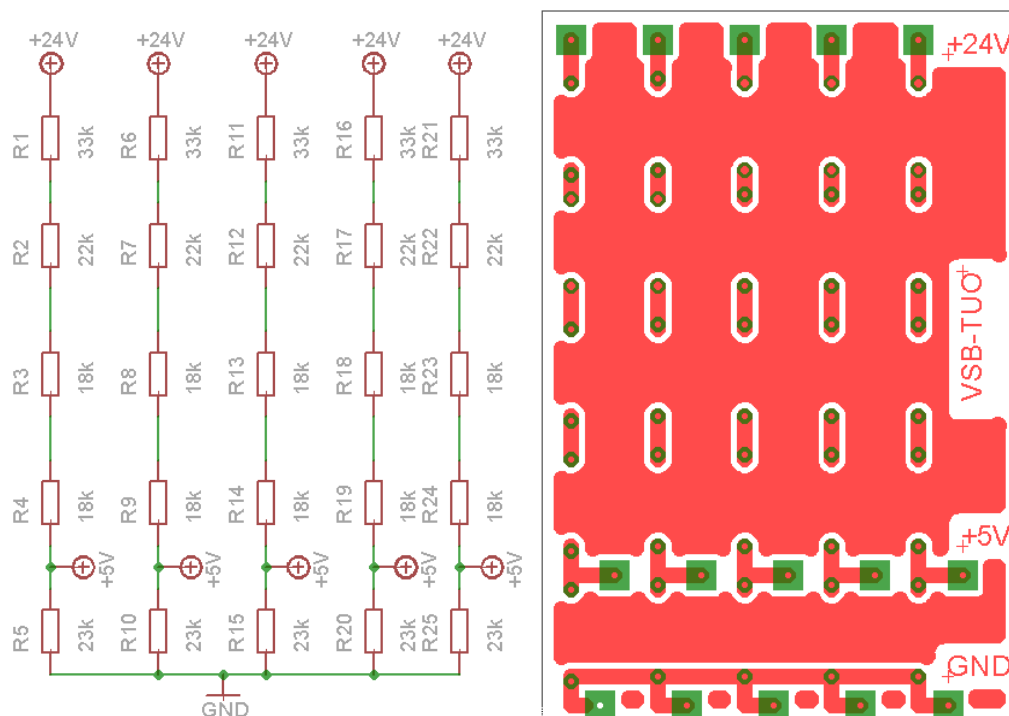
Obr. 12 Přerušená vodivá cesta na desce plošného spoje

4 Připojení desky Arduino k panelu

Jak je již popsáno výše, Arduino má 5V vstupy a výstupy a panel používá průmyslové bezpečné napětí 24V. Plošné desky, které jsou napojeny na relé, již mají pomocí napěťového děliče, popř. ovládacího relé potřebné 5V vstupy a výstupy pro mikroprocesor.

Tlačítka na panelu tuto úpravu neměly a bylo je nutno dodělat. V softwaru Eagle se navrhla deska plošného spoje (Eagle Průvodce, c1994-2014), která obsahuje pět napěťových děličů, jejichž hodnoty odporů jsou naprosto stejné, jako na deskách, které ovládají relé.

Na panelu najdeme tři tlačítka. Dva děliče jsou tedy jako rezerva pro možné budoucí rozšíření.



Obr. 13 Rozšiřující napěťový dělič

Po tomto kroku již byly všechny vstupy a výstupy připraveny k použití. Provedlo se tedy jejich připojení k vývojové desce Arduino. Připojení jednotlivých pinů zobrazuje následující tabulka:

Tab. 3 Přiřazení pinů k jednotlivým relé

Prvky panelu (viz obr. 2, 3, 5, 6)	Piny mikroprocesoru	
	Indikace	Sepnutí
Relé K1	Pin 44	Pin 32
Relé K2	Pin 22	Pin 34
Relé K3	Pin 24	Pin 36
Relé K4	Pin 26	Pin 38
Relé K5	Pin 28	Pin 40
Relé K6	Pin 30	Pin 42
Časové relé (S) - 1	Pin 37	Pin 29
Časové relé (S) - 2	Pin 23	Pin 31
Časové relé (M) - 1	Pin 25	Pin 33
Časové relé (M) - 2	Pin 27	Pin 35
Tlačítko T1 Bez aretace	Pin 45	
Tlačítko T2 S aretací	Pin 47	
Tlačítko T3 S aretací	Pin 49	

Základní algoritmus

Arduino v tomto projektu tvoří jen určitý most mezi panelem a počítačem. Algoritmus v procesoru tedy není moc složitý. Ve funkci *setup* se přiřadily jednotlivým použitým pinům, zdali mají fungovat jako digitální vstupy či digitální výstupy. Dále byly vytvořeny proměnné, ze kterých je zřejmé, o jaký prvek na panelu se starají a jestli se jedná o vstup, nebo výstup. Tyto proměnné se poté inicializovaly k jednotlivým fyzickým pinům na desce Arduina.

Příklad:

```
const int switchPin_O_K1 = 32;
pinMode(switchPin_O_K1, OUTPUT);
const int switchPin_I_K3 = 24;
pinMode(switchPin_I_K3, INPUT);
```

Software, ve kterém se Arduino programuje, obsahuje sériový monitor, díky kterému bylo možno rychle ověřit, jestli bylo vše připojeno správně a jestli vstupy a výstupy fungují, tak jak mají.

Výpis logických stavů jednotlivých vstupů byl umožněn pomocí funkce *Serial.Print*. Naopak ovládání vstupů bylo vyřešeno zadáním hodnoty do sériového monitoru a použitím funkce *Serial.Read*.

```
switchPin_O_K1 = Serial.read();
Serial.print(switchPin_I_K3);
```

5 PROMOTIC

Software PROMOTIC je souhrnný SCADA objektový prostředek pro tvorbu programů, které řídí, sledují a zobrazují technologické procesy v různých oblastech průmyslu.

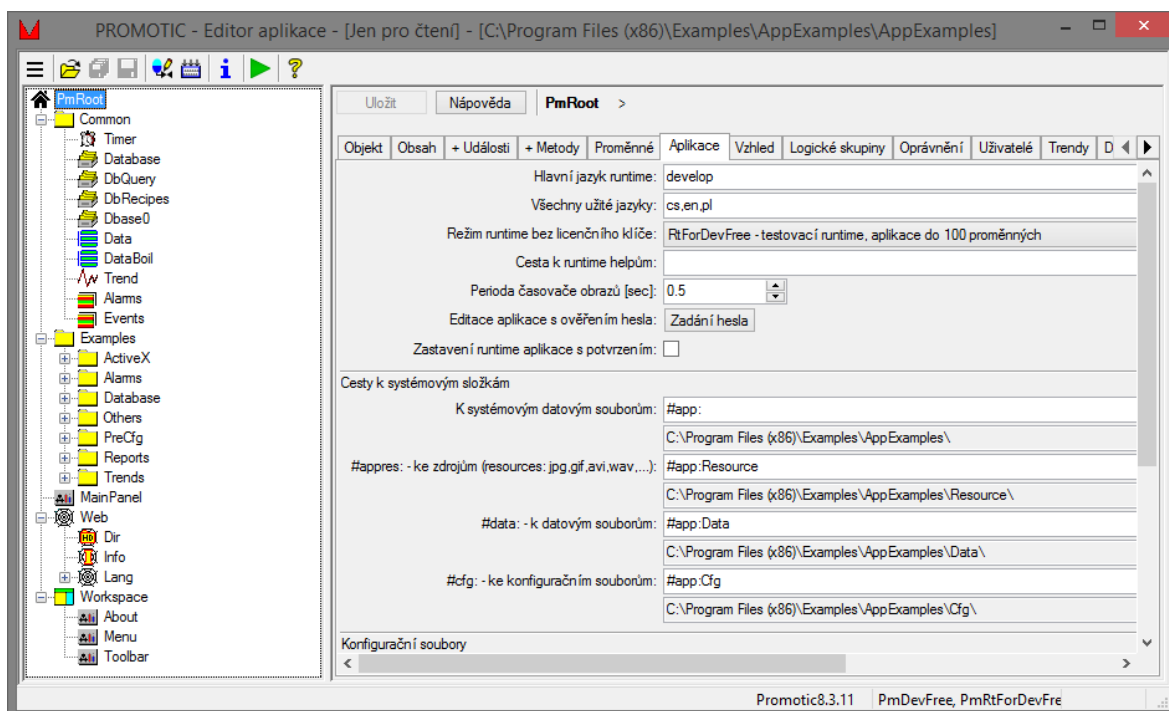
Je navržen pro operační systém Windows XP, 2003 Server a vyšší. Od verze 8 je možné PROMOTIC provozovat také ve freeware módu pro nekomerční účely, která je omezena maximálním počtem obrazů a velikostí aplikace (počet proměnných). Najdeme zde všechny potřebné komponenty pro tvorbu jak jednoduchých řídicích vizualizačních systému, tak i rozsáhlých.

Tento SCADA systém disponuje širokou paletu komunikačních rozhraní (XLM, OPC, ActiveX, DDE,...). K dispozici je také obrovská škála komunikačních ovladačů pro PLC automaty. Mezi podporované typy databází patří MySQL, Access, Dbase, Oracle, SQL, ODBC a další. Jako základní nástroje PROMOTICu je zde Editor aplikace a Editor obrázků. Obsahuje velmi obsáhlou knihovnu grafických prvků a technologických obrázků vytvořených ve vektorové SVG grafice. Tuto knihovnu je možno doplnit vlastními obrázky. Samozřejmostí je systém trendů, alarmů a operátorských událostí. Dále podporuje také web technologii Internet/Intranet, kdy na web serveru máme aplikaci, která poskytuje webové stránky, které si můžeme prohlížet pomocí standardních webových prohlížečů (IE, Firefox, Chrome,...). Existuje několik jazykových verzí softwaru PROMOTIC. Vytvořené aplikace je možno zabezpečit pomocí přihlašovacího systému, či správou uživatelů a jejich oprávnění.

Editor aplikace je hlavní nástroj, díky kterému tvoříme aplikace v systému PROMOTIC. Je používán k definování stromové struktury objektů, jejich nastavení, definování algoritmů apod.

Pro informace a diagnostiku je dostupný INFO systém. Umožňuje vidět všechny důležité a podstatné informace během spuštěné aplikace. Je zde i možnost dálkového odladění aplikace přes internetovou/intranetovou síť přes PROMOTIC Web, případně softwarem TeamViewer, LapLink, PCAnyWhere atd.

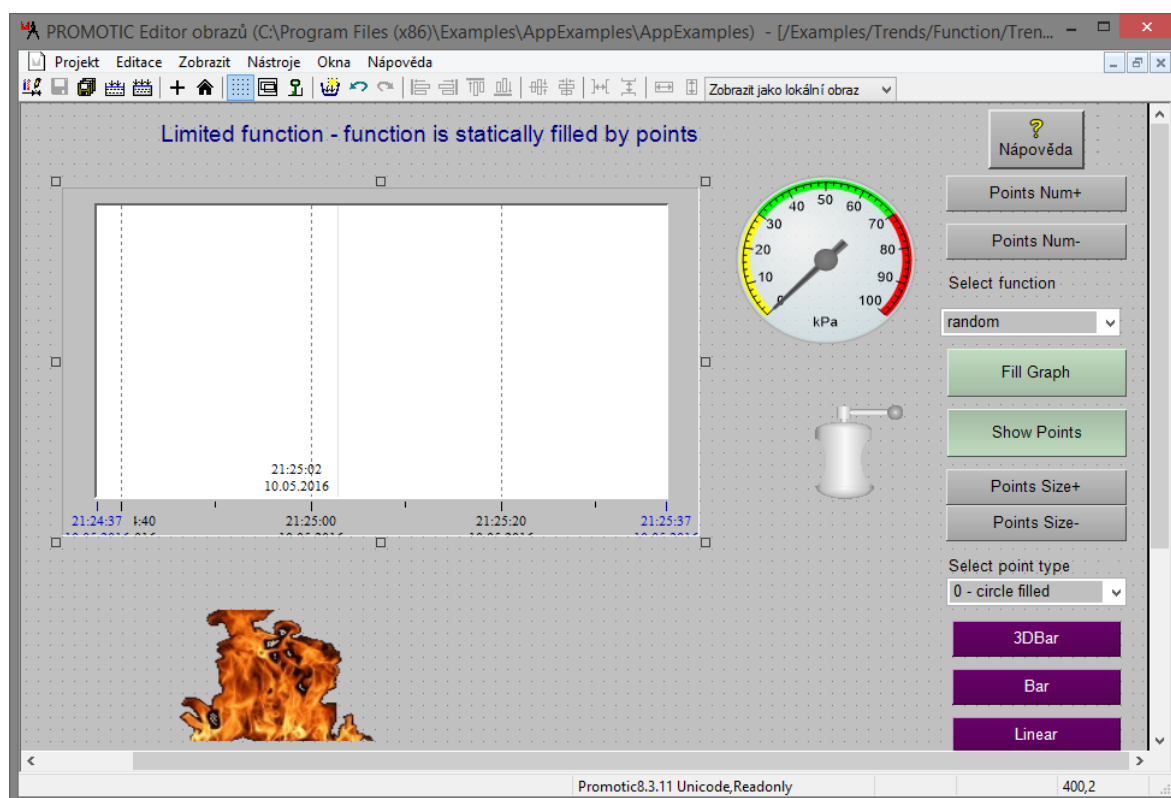
K událostnímu programování slouží pro zápis uživatelských algoritmů zabudovaný jazyk VBScript se syntaxí Visual Basic. Pro vývojáře to symbolizuje neomezené možnosti vývoje.



Obr. 14 PROMOTIC - editor aplikace

Editor obrázků je používán pro vytváření grafických panelů aplikace a to použitím grafických prvků vytvořených designerem podle jeho představ, popř. výběrem předdefinovaných prvků z dostupné palety, či nahraných uživatelských obrázků. V těchto prvcích je možnost nastavit jejich vlastnosti, propojit s datovou vazbou, a tím zprovoznit vizualizaci.

Uživatel může vytvořit panely, které jsou standardními Windows okny s fixní pozicí, velikostí, velikostí přes celou obrazovku, okna bez lišt nebo definovat maximální počet otevřených oken najednou.



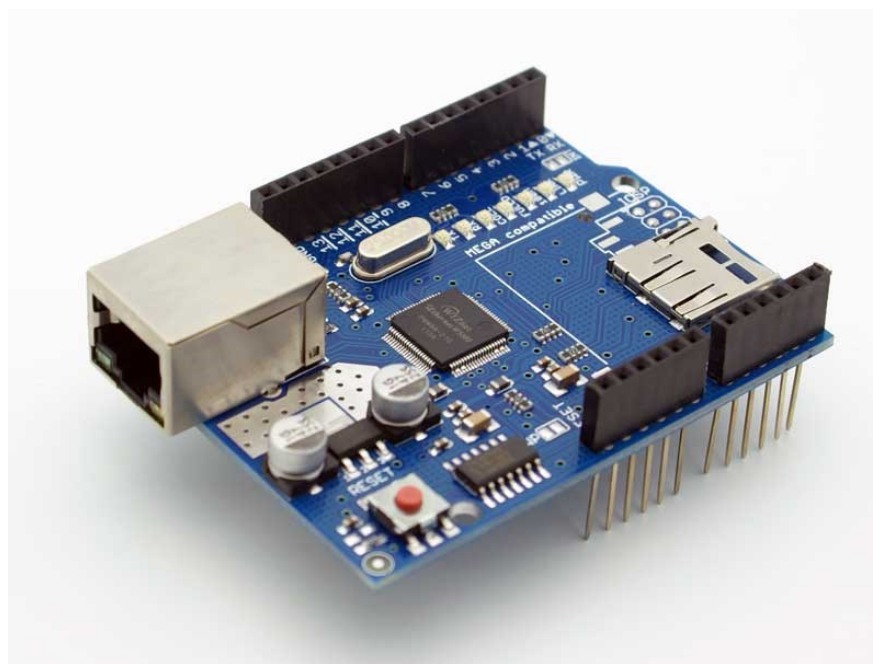
Obr. 15 PROMOTIC - editor obrázků

6 Propojení Arduino – PROMOTIC

Mikrokontrolér Arduino má nesčetně mnoho možností, jak komunikovat s jinými mikrokontroléry a počítači. Pro komunikaci mezi mikrokontrolérem Arduino a softwarem PROMOTIC, respektive počítačem je ideální otevřený komunikační protokol Modbus. PROMOTIC má již integrované ovladače tohoto protokolu a k Arduino jsou volně stažitelné knihovny, které nám přenos umožní. Jde o komunikační protokol na úrovni aplikační vrstvy ISO/OSI modelu vytvořený firmou Modicon (nyní Schneider Electric) v roce 1979 pro vzájemnou komunikaci logických programovatelných automatů. Stal se de facto standardním komunikačním protokolem a nyní je často dostupný pro připojení průmyslových elektronických zařízení. Modbus povoluje komunikace mezi mnoha zařízeními připojenými ve stejné síti (například měření teploty a vlhkosti a posílání výsledků do počítače). Používá se taky často pro SCADA systémy a RTU.

Mezi přenosová média standardně patří síť Ethernet (TCP/IP), sériová linka (RS-232, RS-485, RS-422, optické a rádiové sítě).

Komunikace funguje na principu klient-server neboli master-slave. Na internetu jsou k dispozici již hotové knihovny pro Arduino (Modbus, Modbus TCP), které usnadní práci s komunikací. Stačí tedy knihovny stáhnout a poté ji v programovacím softwaru přidat. Arduino bude z hlediska komunikace na pozici slave a PROMOTIC master. Komunikace je možná buď přes sériovou linku, nebo Ethernet. Ethernetové rozhraní pro Arduino se dá běžně koupit a k desce Mega 2560 lehce připojit. Máme možnost pořídit originální Arduino shield anebo klony vyrobeny jiným výrobcem, které ale fungují naprosto stejně. Knihovna pro ethernetové rozhraní je již v programovacím prostředí implementována.

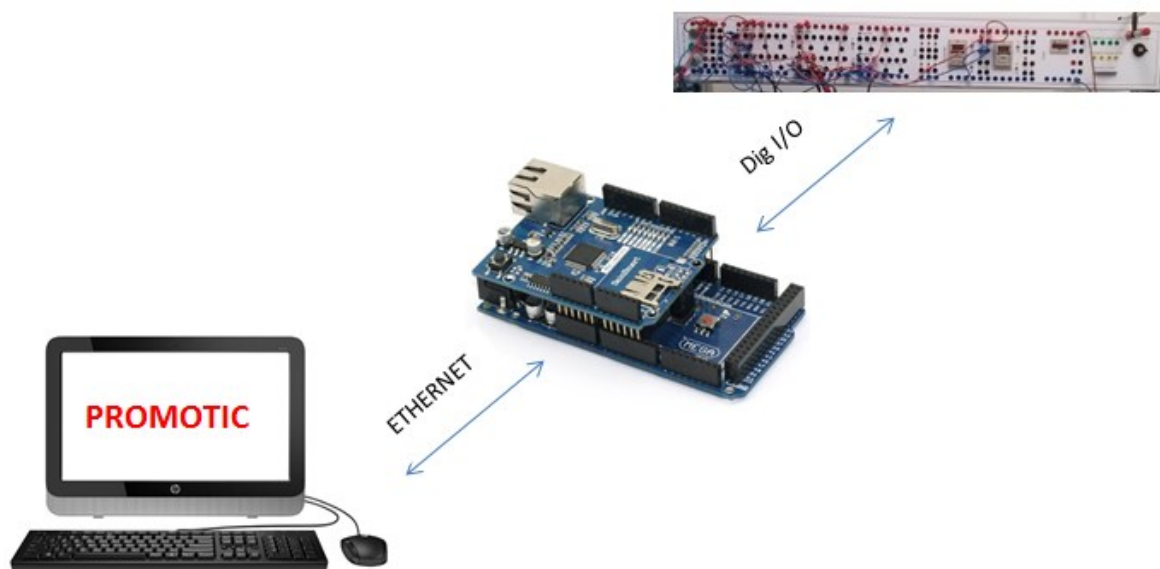


Obr. 16 Ethernetové rozhraní Arduino (ROBOMART, 2015)

Co se týče komunikace přes sériovou linku, tak jsou zde dvě možnosti. Buď jsou data vysílána přes USB sběrnici, nebo zde najdeme standardní RX a TX piny (Mega 2560 má 4 sériové linky). Pokud budeme chtít použít komunikační rozhraní RS-232, je nutné použít převodník na TTL logiku, jelikož Arduino pracuje s napětím 5V a RS-232 v počítači s 12V nebo 15V. K tomuto účelu se používá levný a velmi používaný převodník MAX232.

7 Komunikace

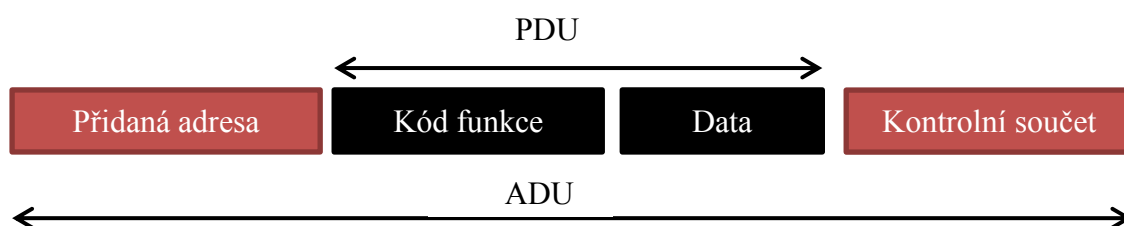
Pro tento konkrétní projekt byla tedy zvolena komunikace pomocí protokolu Modbus a ethernetového rozhraní. Ethernetový kabel je tedy možné připojit do switchu a k mikroprocesoru se poté připojit z kteréhokoliv počítače, který je připojen na stejné síti.



Obr. 17 Schéma zapojení

7.1 Modbus

Modbus protokol definuje jednoduchou protokolovou datovou jednotku (PDU) nezávislou na typu komunikační vrstvy. Mapováním na specifickou sběrnici nebo síť se dále zpráva rozšíří o další část a posune se na úroveň aplikační datové jednotky (APU).



Obr. 18 Princip komunikace Modbus

Kód funkce informuje server, který druh operace má provést. Hodnota je zakódována v jednom bytu (1-255), kdy rozsah 128-255 je vyhrazen pro oznámení chyby. Funkční kód „0“ není platný.

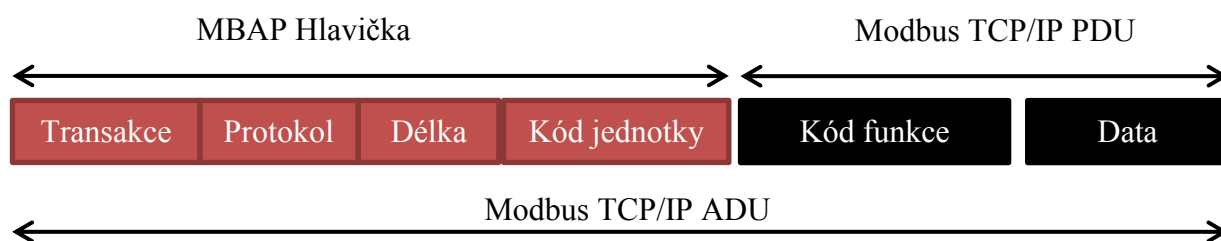
Datové pole zpráv vyslané z klienta na server obsahuje přidané informace, které server použije, aby provedl akci definovanou funkčním kódem. Díky tomu mohou být ve zprávě položky jako diskrétní adresy a adresy registrů, počet položek k zpracování a počet aktuálních datových bytů v poli. U některých druhů funkcí nemusí datové pole existovat, v tomto případě server další informace nevyžaduje.

Při přenosu transakce mohou nastat dvě možnosti. První je, že transakce proběhne provedením požadavku bez chyby. Klient odešle kód funkce a data požadavku. Server tuto zprávu přijme a po úspěšném zpracování odesílá zpět kód provedené funkce a datovou složku odpovědi. V opačném případě (pokud dojde k chybě) vrací server originální kód funkce s nastaveným nejvíce významovým bitem na log. 1. Místo datové odpovědi zpráva obsahuje kód chyby, který nám upřesní její důvod.

Maximální velikost zprávy je zděděna již od první implementace Modbusu na sériové lince (max RS 485 ADU = 256 bytů).

Modbus PDU pro komunikaci po sérové lince má tedy velikost 256 bytů, kdy jeden byte zabírá adresu serveru, a další 2 byty jsou kontrolní součet.

TCP/IP Modbus, který je použit v této úloze má trochu odlišnou formu. Základem je také 253 bytů PDU, plus obsahuje takzvanou MBAP (Modbus Application Protocol Header) hlavičku o délce 7 bytů – celkem tedy 260 bytů. Ta obsahuje pořadové číslo transakce, číslo protokolu, informaci o celkovém počtu bytů a číslo jednotky.



Obr. 19 Modbus TCP/IP

7.2 Datový model

Datový model protokolu Modbus je založen na sérii tabulek, které mají svůj charakteristický význam. Definovány jsou čtyři základní (Modbus, 2006):

Tab. 4 Datové tabulky Modbus

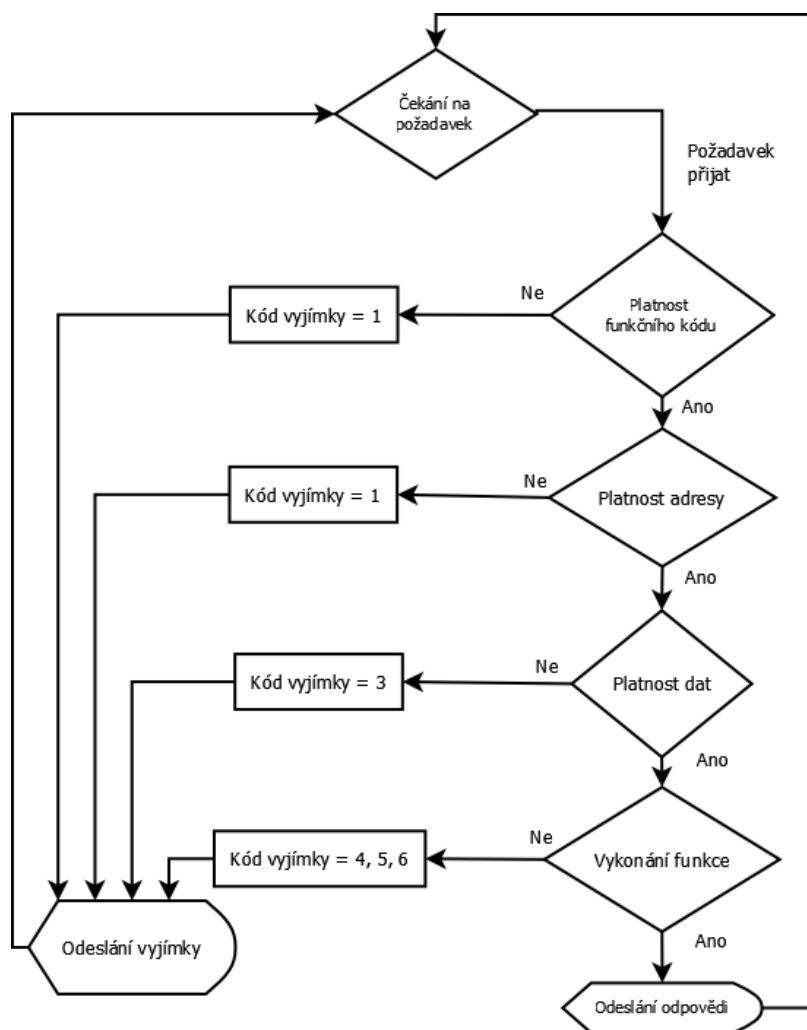
Tabulka	Typ	Přístup	Adresa
Diskrétní vstupy Discrete inputs	1bit	Jen čtení	10001÷19999
Cívky Coils	1bit	Čtení/zápis	1÷9999
Vstupní registry Input registers	16bit	Jen čtení	30001÷39999
Uchovávací registry Holding registers	16bit	Čtení zápis	40001÷49999

Každá tabulka může mít standardně 9998 položek, což nám v tomto projektu stačí bohatě (jen pár desítek položek). Jsou ale případy, kdy jich je potřeba více. Toto řeší novější standard protokolu Modbus, kdy může tabulka obsahovat až 65535 položek. Takto použitá adresa může mít např. hodnotu 165535, kdy nejvyšší řád určuje typ registru a další číslovky jeho hodnotu. Tato notace byla vyvinuta ještě původní firmou Modicon (nyní Schneider Electric). (Modbus 101, 2016).

Co se týče konkrétní transakce, tak Modbus definuje tři datové jednotky:

- Modbus žádost
 - Funkční kód – 1 byte
 - Datová část (počet proměnných, adresa,...) – n bytů
- Modbus odpověď
 - Funkční kód – 1 byte
 - Datová část (přečtené vstupy, počet proměnných, adresy,...) – n bytů
- Modbus výjimka
 - Funkční kód + 0x80 – 1 byte
 - Chybový kód – 1 byt

Následující diagram zobrazuje obecné zpracování Modbus transakce na straně serveru:



Obr. 20 Modbus transakce

Jakmile je žádost serverem zpracována, Modbus vytvoří adekvátní odpověď a odešle ji druhé straně. V závislosti na výsledku zpracování můžeme dostat dva typy odpovědi:

- Pozitivní odpověď:
 - Kód funkce odpovědi = kód funkce požadavku
- Záporná odpověď (výjimka):
 - Cílem je poskytnout relevantní informace, z jakého důvodu chyba nastala
 - Kód funkce odpovědi = kód funkce požadavku + 0x80h

Modbus má tři kategorie funkčních kódů:

Veřejné funkční kódy

- Garance unikátnosti.
- Jasně definované, veřejně dokumentované.
- Uznané Modbus.org komunitou.

Uživatelsky definované funkční kódy

- Jsou zde dva rozsahy uživatelsky funkčních kódů: 65-72 a 100-110.
- Uživatel může implementovat a vybrat funkční kód, který není specifikací podporován.
- Není zde garantováno, že použitý funkční kód bude unikátní.

Rezervované funkční kódy

- Jde o funkční kódy, které jsou v současné době používány firmami pro jejich produkty a nejsou dostupné pro veřejné použití.

Mezi základní funkční kódy patří:

- 0x01 Čti cívky
- 0x02 Čti diskrétní vstupy
- 0x03 Čti uchovávající registry
- 0x04 Čti vstupní registry
- 0x05 Zapiš jednu cívku
- 0x06 Zapiš jeden registr

7.3 Adresace

Nyní bylo potřeba všem zvoleným proměnným, které jsou potřeba pro panel určit jejich adresy.

Tab. 5 Použité proměnné

Název proměnné	Prvek panelu (viz obr. 2, 3, 5, 6)	Adresa
O_K1	Ovládání relé K1	106
O_K2	Ovládání relé K2	107
O_K3	Ovládání relé K3	108
O_K4	Ovládání relé K4	109
O_K5	Ovládání relé K5	110
O_K6	Ovládání relé K6	111
O_CR1	Ovládání čas. relé 1, okruh 1	116
O_CR2	Ovládání čas. relé 1, okruh 2	117
O_CR3	Ovládání čas. relé 2, okruh 1	118
O_CR4	Ovládání čas. relé 2, okruh 3	119
I_K1	Indikace relé K1	100
I_K2	Indikace relé K2	101
I_K3	Indikace relé K3	102
I_K4	Indikace relé K4	103
I_K5	Indikace relé K5	104
I_K6	Indikace relé K6	105
I_CR1	Indikace čas. relé 1, okruh 1	112
I_CR2	Indikace čas. relé 1, okruh 2	113
I_CR3	Indikace čas. relé 2, okruh 1	114
I_CR4	Indikace čas. relé 2, okruh 2	115
T_TL1	Indikace tlačítka TL1	120
T_TL2	Indikace tlačítka TL1	121
T_TL3	Indikace tlačítka TL1	122

Tyto adresy se dále nastavily jak v programu Arduina, tak v ovladači Modbus PROMOTIC.

7.4 Nastavení komunikace v Arduinu

Jako první je nutno aktivovat potřebné knihovny pro komunikaci. Potřebujeme *Ethernet.h*, *Modbus.h* a jelikož komunikuje přes TCP/IP tak rozšíření *ModbusIP.h*. Knihovnu *Ethernet.h* již programovací software standardně obsahuje, zbylé dvě bylo potřeba doinstalovat.

Dále bylo potřeba ethernetovému rozhraní přiřadit MAC a IP adresu. Toto nastavení pak dále předat i Modbus knihovně.

```
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };  
  
byte ip[] = { 192, 168, 1, 120 };  
  
mb.config(mac, ip);
```

Nastavení vstupů/výstupů již bylo popsáno výše. Nyní už stačí vytvořeným proměnným přiřadit adresy podle tabulky a přiřadit funkci z Modbus knihovny podle toho, zdali se jedná o vstup anebo výstup.

Kompletní obsluha výstupu vypadá takto:

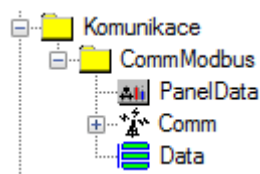
```
const int SWITCH_ISTS_O_K3 = 108; //adresa proměnné  
  
const int switchPin_O_K3 = 36; //ovládání relé K3 - pin 36  
  
pinMode(switchPin_O_K3, OUTPUT); //nastavení pinu - výstup  
  
mb.addCoil(SWITCH_ISTS_O_K3); //funkce addCoil pro DO  
  
digitalWrite(switchPin_O_K3, mb.Coil(SWITCH_ISTS_O_K3));
```

Kompletní obsluha vstupu vypadá takto:

```
const int SWITCH_ISTS_I_TL1 = 120;  
  
const int switchPin_I_TL1 = 45;  
  
pinMode(switchPin_I_TL1, INPUT);  
  
mb.addIsts(SWITCH_ISTS_I_TL1);  
  
mb.Ists(SWITCH_ISTS_I_TL1, digitalRead(switchPin_I_TL1));
```

7.5 Nastavení komunikace v softwaru PROMOTIC

Nastavení Modbus komunikace v prostředí PROMOTIC je jednodušší. V editoru aplikace a kořenové položce PmRoot byl přidán ovladač PmModbusMr. Vytvořila se složka, kde se nachází veškeré potřebné nastavení.



Obr. 21 Vložený ovladač protokolu Modbus

V parametrech komunikace byla nastavena adresa mikroprocesoru a další důležité vlastnosti ethernetu.

Parametry Ethernet:	
192.168.1.120	Sít'ová adresa
502	Číslo TCP/UDP portu
TCP	Typ Ethernet přenosu
15	Počet opakování při neúspěšném přenosu
2000	Timeout přenosu jednoho paketu [ms]
Ne	Filtrovat ECHO znaky
Ne	Připojit až při prvním přenosu
Ano	Uzavřít spojení po každém přenosu
1000	Max.počet řádků v INFO/COMM, záložce 'Monitor'

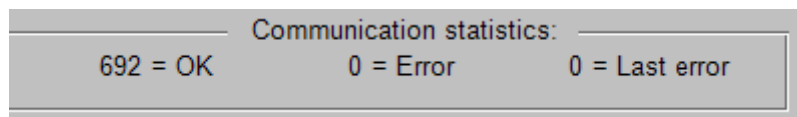
Obr. 22 Parametry ethernetu

Do položky *Data* byly vloženy všechny proměnné a jejich adresy podle tabulky č. 5.

0	O_K1	Boolean	digital output	comm={Comm:/Comm/Data;devD.Bo106;1;0;1}
1	O_K2	Boolean	digital output	comm={Comm:/Comm/Data;devD.Bo107;1;0;1}
2	O_K3	Boolean	digital output	comm={Comm:/Comm/Data;devD.Bo108;1;0;1}
3	O_K4	Boolean	digital output	comm={Comm:/Comm/Data;devD.Bo109;1;0;1}
4	O_K5	Boolean	digital output	comm={Comm:/Comm/Data;devD.Bo110;1;0;1}
5	O_K6	Boolean	digital output	comm={Comm:/Comm/Data;devD.Bo111;1;0;1}
6	O_CR1	Boolean	digital output	comm={Comm:/Comm/Data;devD.Bo116;1;0;1}
7	O_CR2	Boolean	digital output	comm={Comm:/Comm/Data;devD.Bo117;1;0;1}
8	O_CR3	Boolean	digital output	comm={Comm:/Comm/Data;devD.Bo118;1;0;1}
9	O_CR4	Boolean	digital output	comm={Comm:/Comm/Data;devD.Bo119;1;0;1}
10	I_K1	Boolean	digital input	comm={Comm:/Comm/Data;devD.Bi100;1;0;1}
11	I_K2	Boolean	digital input	comm={Comm:/Comm/Data;devD.Bi101;1;0;1}
12	I_K3	Boolean	digital input	comm={Comm:/Comm/Data;devD.Bi102;1;0;1}
13	I_K4	Boolean	digital input	comm={Comm:/Comm/Data;devD.Bi103;1;0;1}
14	I_K5	Boolean	digital input	comm={Comm:/Comm/Data;devD.Bi104;1;0;1}
15	I_K6	Boolean	digital input	comm={Comm:/Comm/Data;devD.Bi105;1;0;1}
16	I_CR1	Boolean	digital input	comm={Comm:/Comm/Data;devD.Bi112;1;0;1}
17	I_CR2	Boolean	digital input	comm={Comm:/Comm/Data;devD.Bi113;1;0;1}
18	I_CR3	Boolean	digital input	comm={Comm:/Comm/Data;devD.Bi114;1;0;1}
19	I_CR4	Boolean	digital input	comm={Comm:/Comm/Data;devD.Bi115;1;0;1}
20	I_TL1	Boolean	digital input	comm={Comm:/Comm/Data;devD.Bi120;1;0;1}
21	I_TL2	Boolean	digital input	comm={Comm:/Comm/Data;devD.Bi121;1;0;1}
22	I_TL3	Boolean	digital input	comm={Comm:/Comm/Data;devD.Bi122;1;0;1}

Obr. 23 Použité proměnné v komunikaci Modbus

Po tomto kroku jsou již nastaveny všechny důležité parametry a může se komunikace otestovat. Při vkládání ovladače do aplikace se automaticky taky vkládá jednoduchá grafická obrazovka PanelData, ve které si po startu vizualizace můžeme zkontrolovat, zda je komunikace v pořádku anebo obsahuje nějakou chybu.



Obr. 24 Diagnostika komunikace

8 Vizualizační aplikace

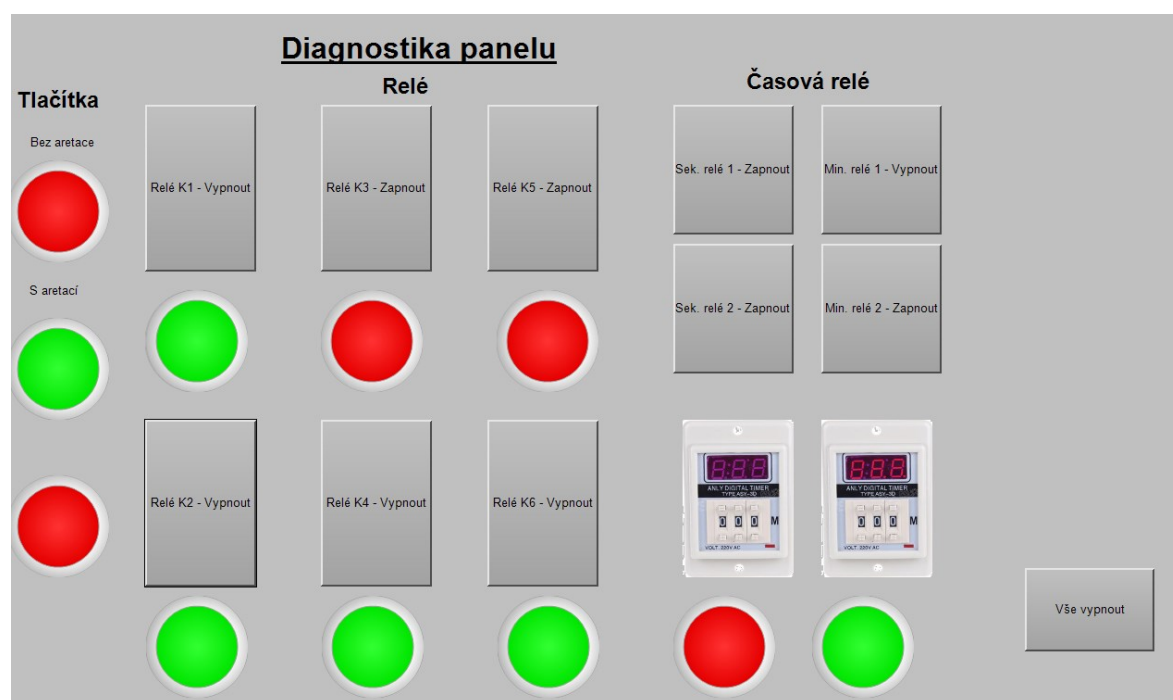
Jelikož je komunikace hotová a funkční, je možno se přesunout do grafického editoru a vytvořit první vizualizační aplikace. V PROMOTICu je na výběr mezi dvěma skriptovacíma jazyky – VisualBasic Script a JavaScript. Pro tento projekt byl zvolen VisualBasic, protože JavaScript musí fungovat i ve WEB obrazech a nejde psát skripty přímo z grafického prvku do aplikačních proměnných.

8.1 Diagnostický panel

Na této obrazovce jsou zobrazeny jednotlivé prvky panelu. Rozmístění prvků v aplikaci odpovídá rozmístění prvků na fyzickém panelu.

Diagnostická obrazovka nám umožňuje spínat pomocí dvoustavových tlačítek jednotlivé relé. Zpětnou vazbou jsou kontrolky, které nám signalizují, zdali k sepnutí opravdu došlo. Dále je možné pomocí signálků monitorovat stavy jednotlivých tlačítek. Každé časové relé má dva okruhy, ale po uplynutí času se sepnou oba výstupy. Z tohoto důvodu postačuje pouze jedna signálka. Dva oddělené okruhy můžeme využít například, pokud chceme relé jednou spouštět přímo z panelu a poté vzdáleně. Díky přepínačům MCU ON/OFF dosáhneme toho, že se jednotlivé okruhy nebudou ovlivňovat.

Zelená signálka zobrazuje sepnuté tlačítko, červená signálka rozepnuté tlačítko/relé.



Obr. 25 Diagnostika panelu

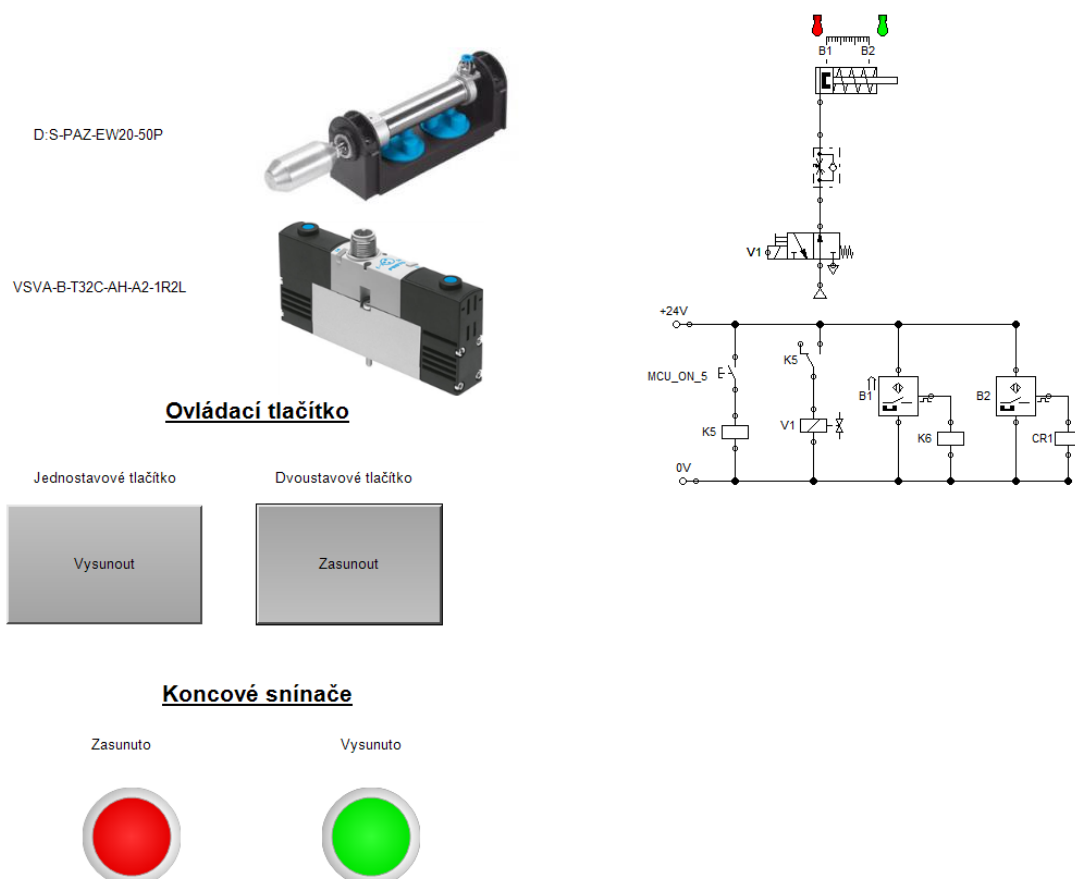
8.2 Jednočinný pneumatický válec

Jako jedna ze základních úloh byla vytvořena úloha, která zobrazuje jednoduchý pneumatický jednočinný válec, jenž je vrácen do základní polohy pružinou. Dává tak možnost začátečníkům se seznámit se základními prvky, které se používají.

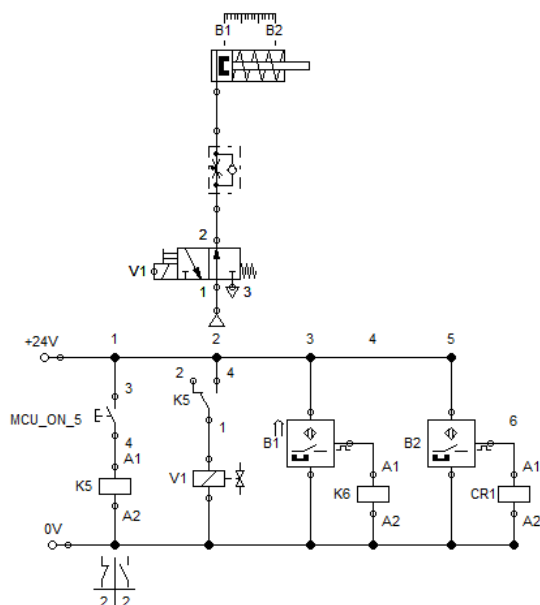
Program zobrazuje použité díly (pneumatický válec, rozvaděč) jak modelovým označením, tak i jeho grafickou podobou. Hned vedle můžeme vidět schéma zapojení úlohy (Festo FluisSIM 4, 2006). Válec můžeme ovládat jak jednostavovým tlačítkem (píst je vysunut jen, pokud je tlačítko stlačeno) anebo dvoustavovým tlačítkem (stačí jen kliknout a píst zůstane vysunutý, pro zasunutí je potřeba opět kliknout).

Pro snímání koncových poloh jsou zde dva koncové snímače a jejich stavy jsou vyobrazeny jak velkými signálkami, tak malými kontrolkami přímo ve schématu zapojení.

Jednočinný válec



Obr. 26 Aplikace s jednočinným válcem



Obr. 27 Schéma zapojení obvodu s jednočinným válcem

8.3 Dvojčinný pneumatický válec

Paralelně s výše uvedenou úlohou byla také vytvořená úloha, která zastupuje funkci dvojčinného pneumatického válce.

Tento válec má oproti jednočinnému zásadní změnu v tom, že obsahuje navíc jeden vstup a nemá svou výchozí pozici po odpojení externího zdroje. Použit je 5/2 rozvaděč, který je ovládaný elektricky, případně pokud je to nutné tak manuálně.

Dvojčinný válec

D-S-PAZ-DW20-100PPV

VSVA-B-B52-H-A2-1R2L

Ovládací tlačítko

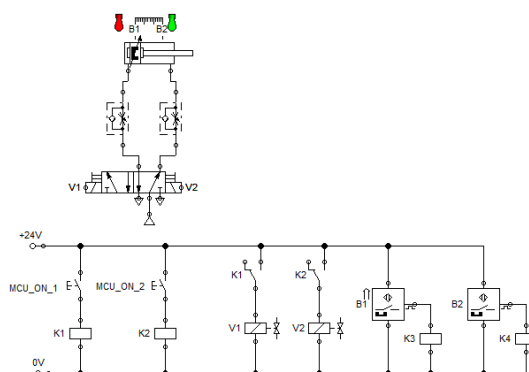
Zasunout

Vysunout

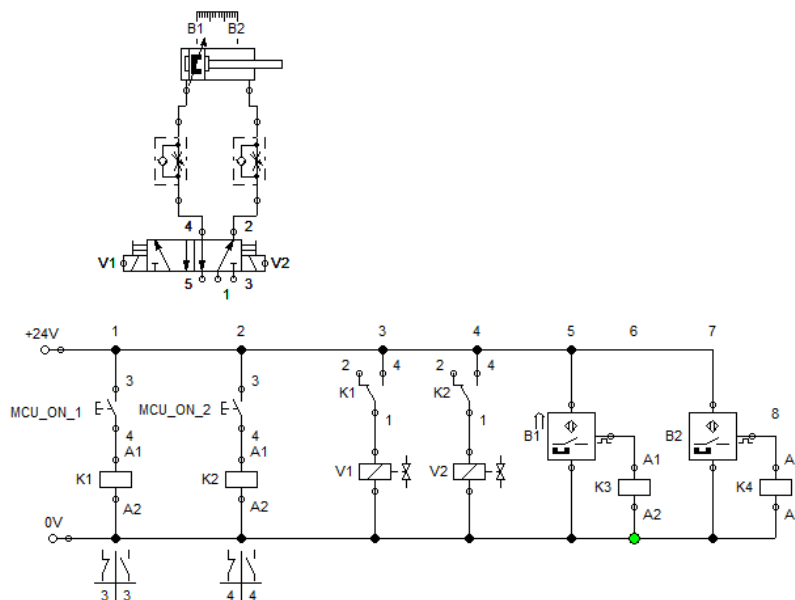
Koncové snínače

Zasunuto

Vysunuto

**Obr. 28 Aplikace s dvojčinným válcem**

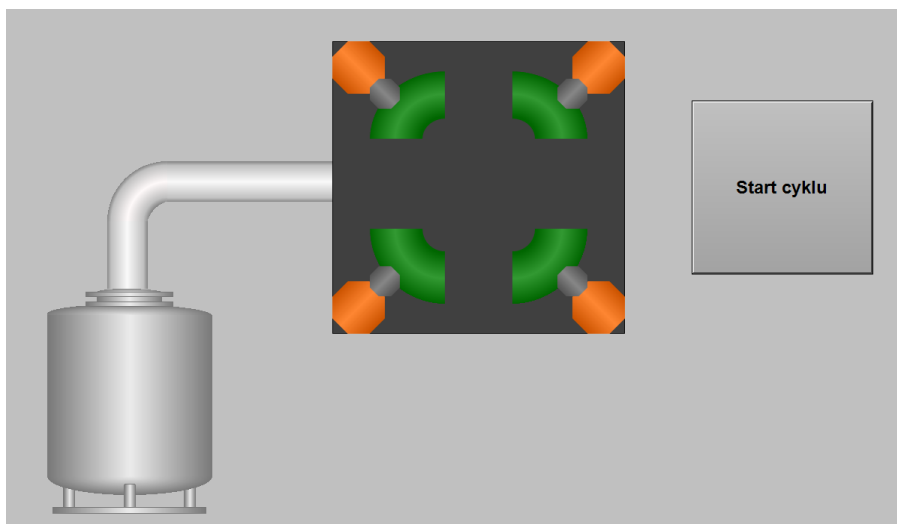
Grafická obrazovka opět zobrazuje základní použité prvky, schéma zapojení, signálky zobrazující koncové polohy pneumatického válce a samozřejmě ovládací tlačítka.

**Obr. 29 Schéma zapojení obvodu s dvojčinným válcem**

8.4 Vizualizace licího stroje

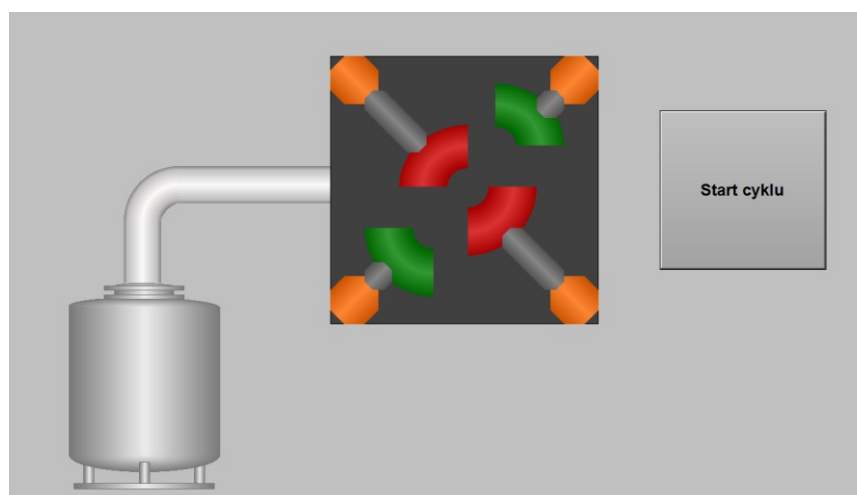
Tato úloha zobrazuje proces výroby hliníkového kola na licím stroji. Jednotlivé kroky jsou řízeny pomocí časovačů, kdy každý krok si hlídá, jestli se předchozí provedl. Není možné tedy spustit např. tlakování nádoby bez toho, aniž by byly zavřené čelisti.

Před startem licího cyklu jsou čelisti otevřeny a pec je odtlakována. Licí cyklus spustíme jednostavovým tlačítkem *Start cyklu*.



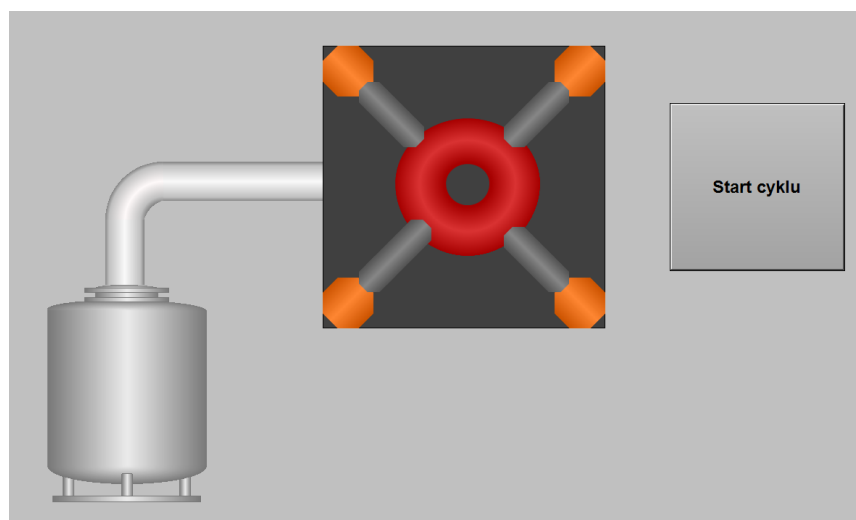
Obr. 30 Základní obrazovka licího stroje

Po spuštění cyklu se zasune čelist 1 a 4 (relé K1).



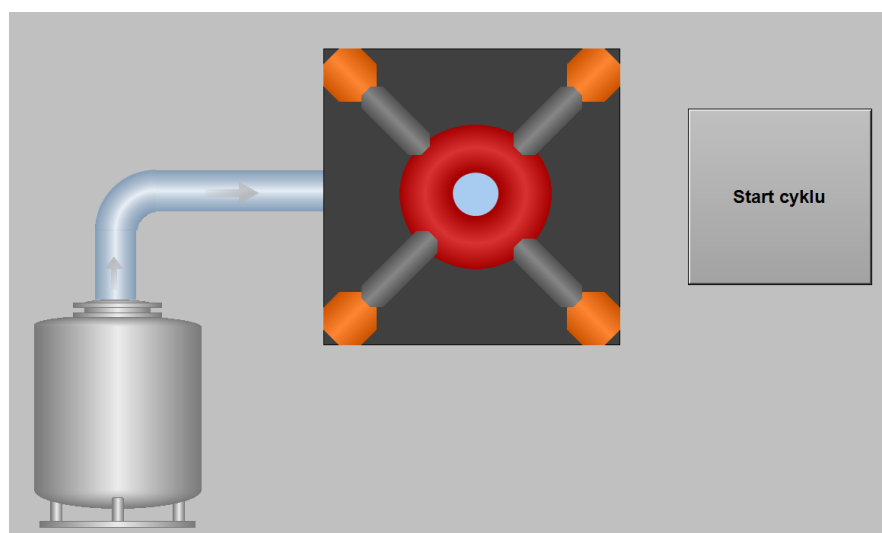
Obr. 31 Zasunutí čelistí 1 a 4

Spolu se zasunutím čelistí se také aktivuje časovač, který po stanoveném čase zasune další dvě čelisti (relé K2).



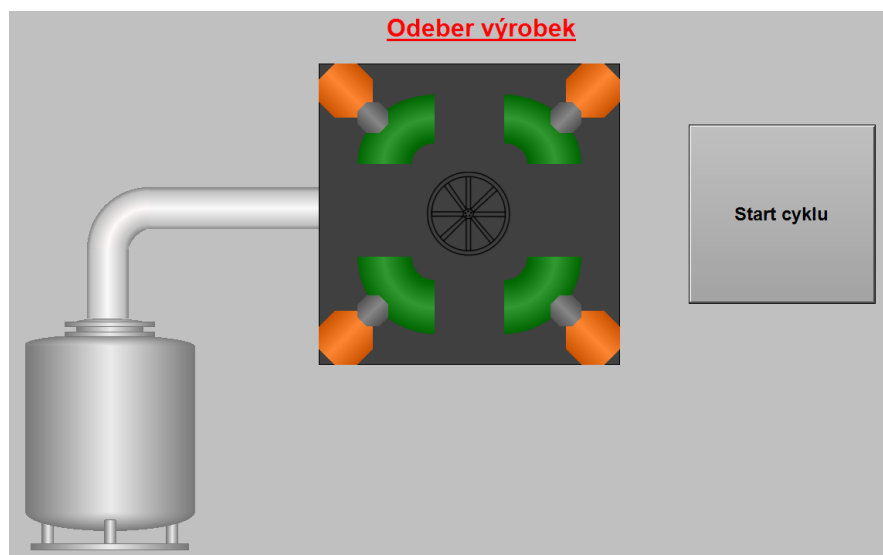
Obr. 32 Zasunutí čelistí 2 a 3

Po zavření všech čelistí se povolí další časovač, který spustí tlakování formy (relé K3).



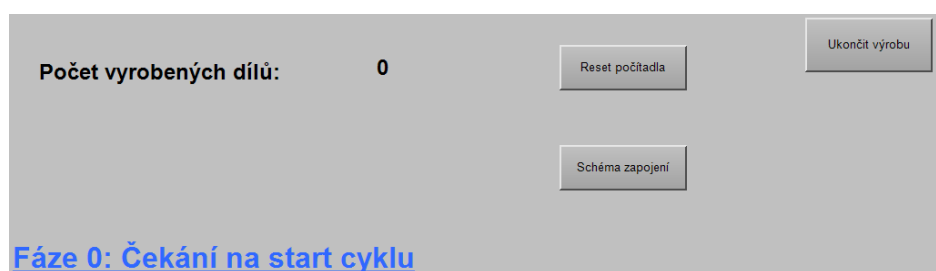
Obr. 33 Tlakování nádoby

Po zvoleném čase se vypne tlakování a začnou se postupně otevírat čelisti (po dvou kusech v úhlopříčce). Na vizualizaci se objeví hotové kolo, které se odebere potvrzovacím tlačítkem TL1.



Obr. 34 Hotový výrobek

Po odebrání kola je možno odstartovat další cyklus.



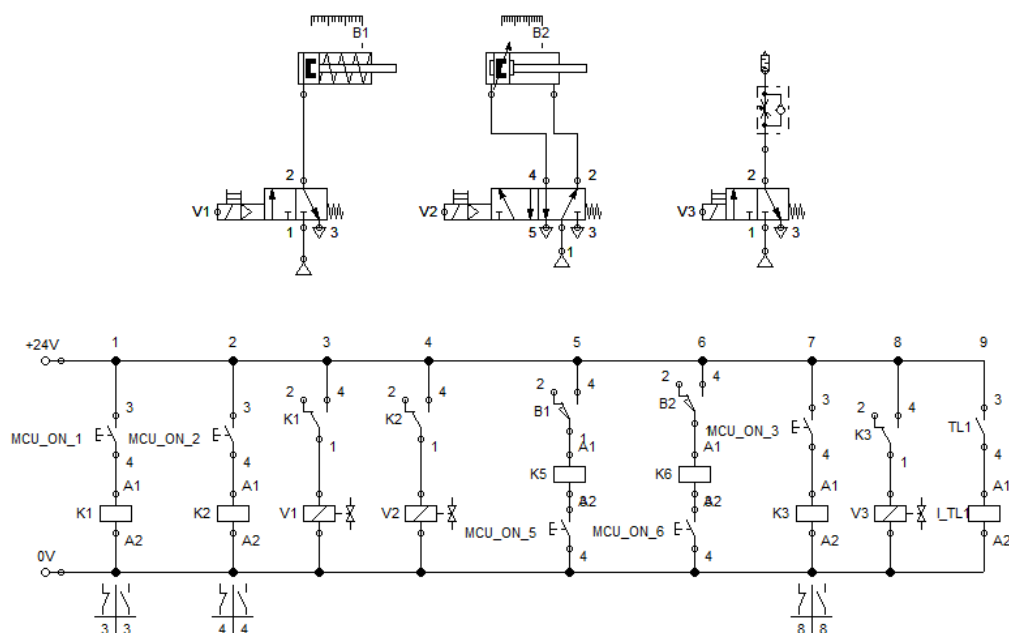
Obr. 35 Menu aplikace Lící stroj

Aplikace dále počítá množství vyrobených dílů od zahájení výroby. Informace o každém vyrobeném dílu se uloží do databáze PROMOTICu (číslo dílu, čas začátku cyklu, čas konce cyklu, délka výrobního procesu v sekundách) a stisknutím tlačítka „Ukončit výrobu“ provedeme uložení těchto informací do souboru Data.csv (umístěný ve složce aplikace), který můžeme jednoduše otevřít např. v softwaru MS Excel.

	A	B	C	D
1	0	13.4.2017 11:07:58	13.4.2017 11:08:16	18
2	1	13.4.2017 11:08:27	13.4.2017 11:08:45	18
3	2	13.4.2017 11:08:52	13.4.2017 11:09:10	18

Obr. 36 Uložené data v souboru

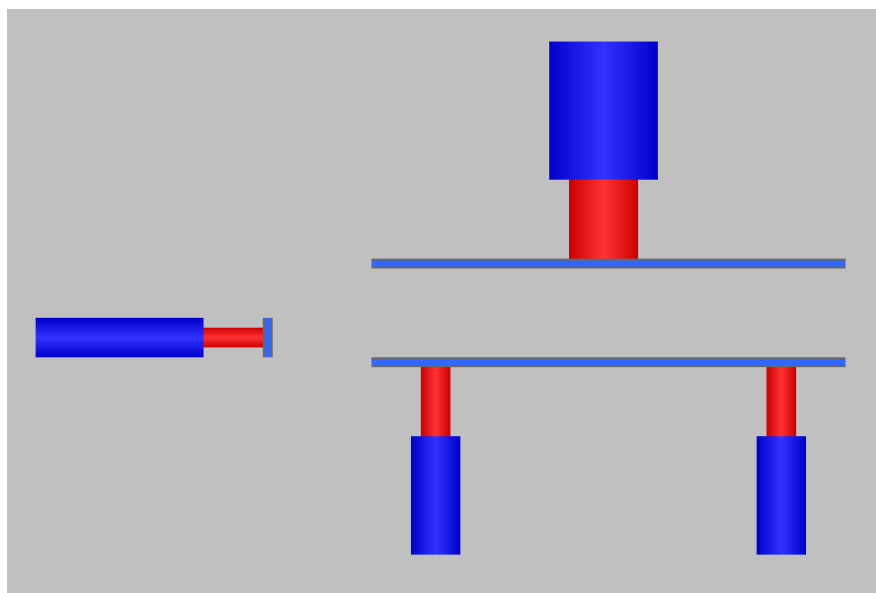
Samozřejmostí je také tlačítko, přičemž jeho stisknutí nám otevře schéma zapojení této úlohy.



Obr. 37 Schéma zapojení úlohy lící stroj

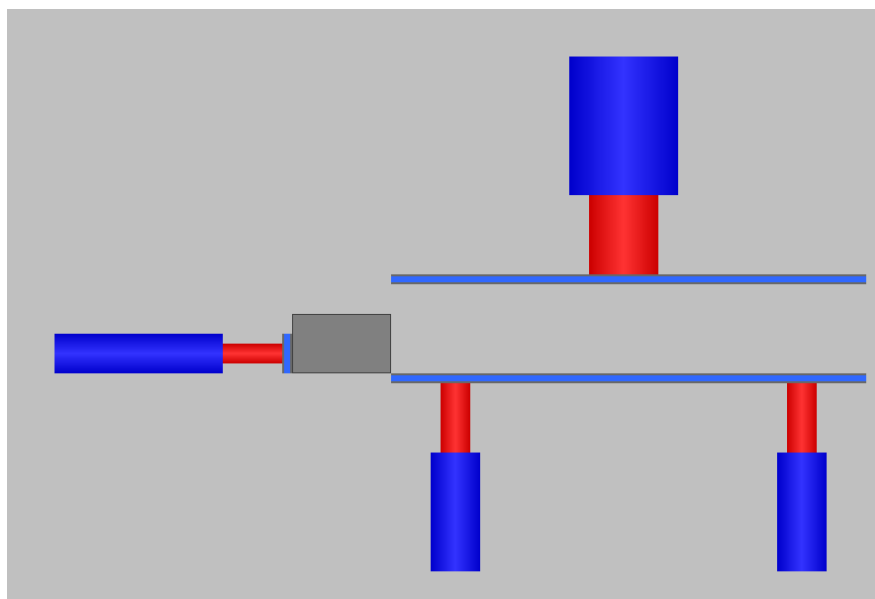
8.5 Vizualizace lisu

Jde o úlohu, která zobrazuje jednoduchý proces lisování polotovaru na finální výrobek.

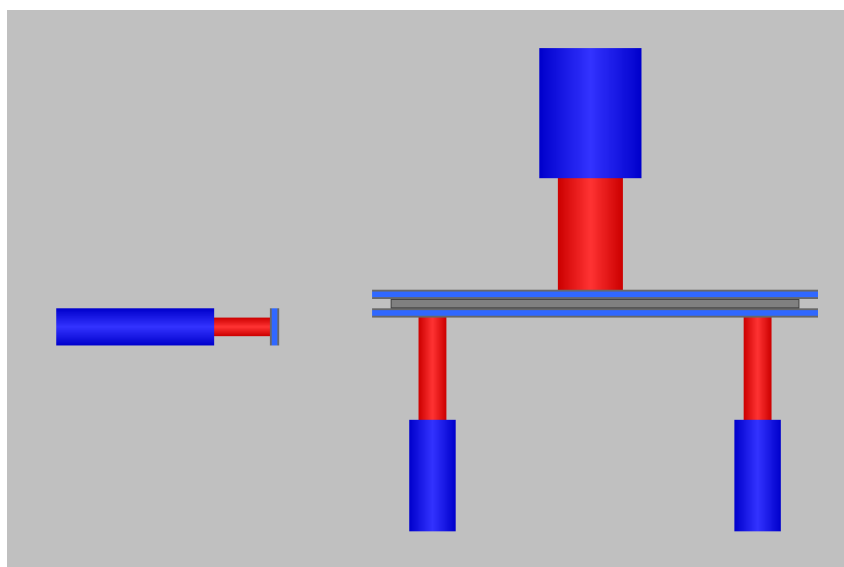


Obr. 38 Základní obrazovka lisu

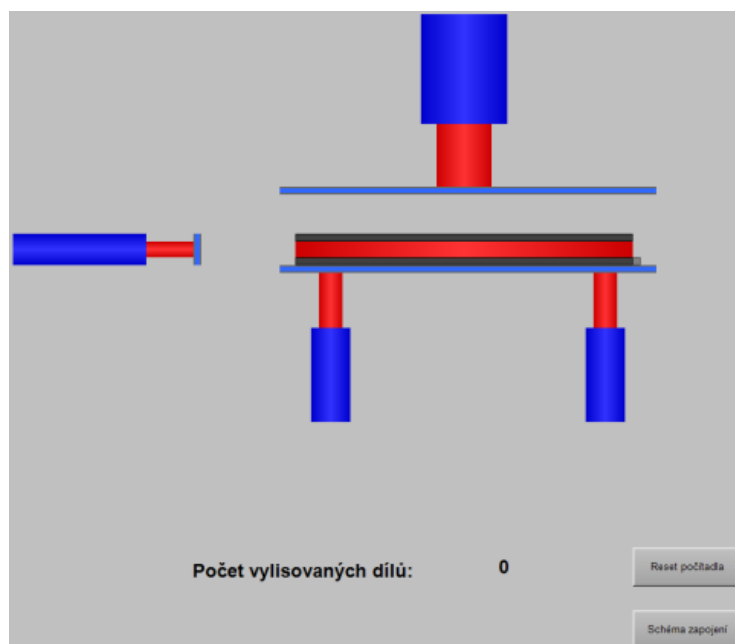
Na začátku se čeká na přítomnost materiálu u podavače.

**Obr. 39 Přítomnost materiálu**

Pokud se materiál u podavače objeví, můžeme tlačítkem TL1 odstartovat cyklus lisování. Podavač zasune materiál a místo pod lis. Po dosáhnutí koncové polohy se vrátí zpět na původní pozici a lis se dá do pohybu.

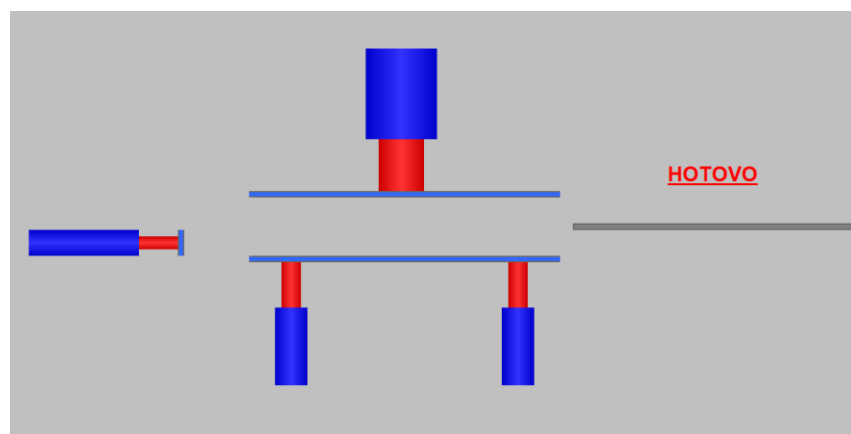
**Obr. 40 Stlačení materiálu**

Po stlačení materiálu se válce opět zasunou.



Obr. 41 Otevření lisu

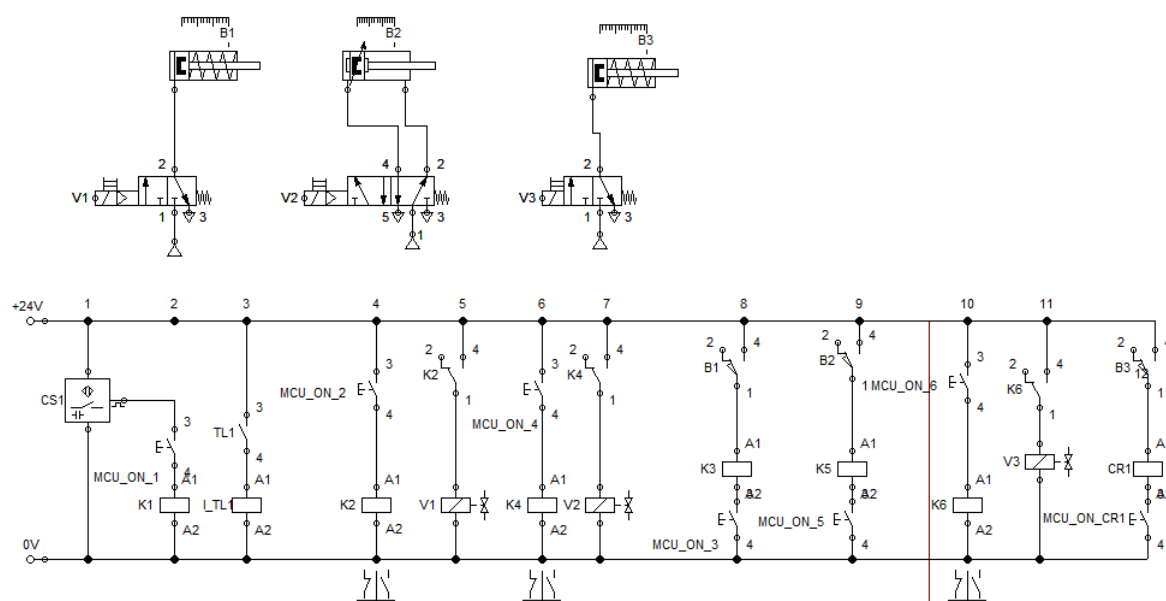
A další válec vytlačí vylisovaný materiál mimo lis.



Obr. 42 Vysunutý materiál mimo lis

Po odebrání materiálu je možno začít s dalším lisováním.

Tato aplikace obsahuje stejné funkce jako předchozí úloha (tzn. čítač zpracovaných dílů, odkaz na schéma zapojení úlohy, zápis informací do souboru).



Obr. 43 Schéma zapojení úlohy lis

8.6 Vizualizace na mobilním zařízení se systémem Android

V současné době se také klade velká míra důležitosti na to, aby bylo vše co možná nejvíce mobilní, přenositelné. Z tohoto důvodu byla také ještě vytvořena mobilní aplikace pro mobilní telefony či tablety se systémem Android.

Vizualizační aplikace opět umožňuje diagnostiku panelu, tzn., můžeme spínat/rozepínat jednotlivé relé a snímat jejich současný stav. Pokud je relé nebo tlačítko sepnuté, rozsvítí se zelená signálka.

Aplikace je vytvořena v softwaru HMI Modbus 1.33, který je volně stažitelný na Google Play. Tato aplikace je primárně určena pro Modbus TCP/IP komunikaci, což je pro tento účel naprosto ideální. Projekt je možno exportovat a přenášet na různá zařízení a poté v tomto softwaru opět spustit.

Pro správnou komunikaci musí být zařízení provozující tuto aplikaci opět připojen na stejné síti jako mikroprocesorová jednotka Arduino.



Obr. 44 Vizualizační aplikace na systému Android

9 Závěr

Cílem diplomové práce bylo seznámit se a popsat výukový elektronický panel pro ovládání a měření stavů elektro-pneumatických prvků, který byl vytvořen v rámci bakalářské práce.

Byl proveden výpis jednotlivých prvků, který tento panel obsahuje, a také byly popsány jejich základní vlastnosti. Pomocí typových úloh se zjistila jejich funkčnost a jakékoliv zjištěné chyby a poruchy byly opraveny (chybný napěťový dělič, vadné časové relé, přerušené cesty na deskách plošných spojů).

Následně byla navržena a vytvořena deska plošných spojů, která obsahuje napěťové děliče. Tyto děliče se použily pro tlačítka na panelu, jelikož neměla 5V výstup pro mikroprocesorovou jednotku.

Dále bylo provedeno připojení mikroprocesorové jednotky, která je využita pro ovládání a monitorování elektronického panelu, popsány její vlastnosti a pomocí jednoduchého algoritmu ověřena funkčnost.

V dalším kroku proběhlo seznámení s vizualizačním prostředím SCADA softwaru PROMOTIC a byly zjištěny ideální možnosti připojení a následná komunikace s mikroprocesorovou jednotkou. Pro komunikaci byl vybrán otevřený protokol Modbus, který komunikuje přes Ethernet TCP/IP. Vývojová deska Arduino byla proto rozšířena o ethernetový modul. PROMOTIC v této komunikaci funguje jako master, Arduino deska jako slave. Po připojení byl v mikroprocesoru vytvořen algoritmus, který ovládá/snímá všechny použité vstupy/výstupy a komunikuje s programem PROMOTIC. Tato deska zde funguje jen jako určitý prostředník mezi panelem a vizualizačním softwarem. Panel byl napájen originálním 24V zdrojem Festo a deska Arduino i s ethernetovým rozšířením pomocí USB rozhraní.

Na závěr byly vytvořeny vizualizační aplikace. První aplikace umožňuje provádět diagnostiku panelu. Z této aplikace je možno ovládat a sledovat jednotlivé prvky umístěné na panelu.

Následně byly paralelně vytvořeny elektro-pneumatické úlohy zobrazující zapojení a funkci jednočinného a dvojčinného pneumotoru, které umožňují seznámit začátečníka se základními elektro-pneumatickými prvky panelu (pneumatické válce, rozvaděče, koncové snímače).

Další dvě aplikace jsou již konkrétní elektro-pneumatické úlohy, které nám zobrazují určitý technologický proces. První aplikace zobrazuje proces výroby hliníkového kola na licím stroji a druhá aplikace zobrazuje lisování polotovaru na finální výrobek.

Závěrem byla ještě vytvořena další diagnostická vizualizační aplikace, která je ale spustitelná na operačním systému Android. Použit byl software HMI Modbus 1.33, který primárně slouží pro Modbus TCP/IP komunikaci.

Ke každé konkrétní úloze byly závěrem v softwaru Festo FluidSIM vytvořeny schémata zapojení, které umožňují celý navržený proces vyzkoušet na laboratorním stole.

Celá tato sestava může být využívána jako nástroj pro studenty do cvičení, kde si můžou vyzkoušet zapojit jednotlivé elektro-pneumatické prvky a ověřit si nově nabyté znalosti. Další možností je rozšíření o další typové vizualizační úlohy v softwaru PROMOTIC, případně rozšíření elektrického panelu o další fyzické prvky, jako jsou analogové vstupy/výstupy apod.

Poděkování

Rád bych poděkoval vedoucímu práce Ing. Mirku Mahdalovi, Ph.D. za poskytnutí tématu, odborné vedení, cenné rady a pomoc při vypracování diplomové práce

10 Použitá literatura

Arduino MEGA 2560. In: Arduino: MEGA 2560 [online]. 2015 [cit. 2016-06-16]. Dostupné z: <https://www.arduino.cc/en/Main/arduinoBoardMega2560>.

Arduino Mega 2560. In: *Pololu robotics & electronics* [online]. US [cit. 2016-06-25]. Dostupné z: <https://www.pololu.com/product/1699>.

Eagle Průvodce. *Eagle Online* [online]. Ochoz u Brna: ELCAD, c1994-2014 [cit. 2016-10-02]. Dostupné z: <http://www.eagle.cz/tour/tour00.htm>.

Festo FluidSIM 4. Festo: Didactic [online]. Denkendorf: Festo, 2006 [cit. 2016-02-02]. Dostupné z: <http://www.festo-didactic.com/ov3/media/customers/1100/00422240001150875008.pdf>.

Festo: Produkty. Festo [online]. Festo, c2017 [cit. 2017-02-02]. Dostupné z: https://www.festo.com/cat/cs_cz/products.

HEROUT, P. 2004. Učebnice jazyka C. 4.vyd. České Budějovice: KOPP a.s., 2004. ISBN: 978-80-7232-383-8.

KOPÁČEK, J. 1996. Pneumatické mechanismy. Díl 1. Pneumatické prvky a systémy. Skripta VŠB-TU Ostrava, 1996.

KOPÁČEK, J. 2005. Pneumatické mechanismy. Díl 2. Řízení pneumatických systémů. 2005. Skripta VŠB-TU Ostrava, 2005.

MODBUS APPLICATION PROTOCOL SPECIFICATION. *Modbus* [online]. 2006 [cit. 2017-01-20]. Dostupné z: http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b.pdf.

Modbus 101 - Introduction to Modbus. *Control solutions* [online]. Minesota, c2016 [cit. 2017-05-10]. Dostupné z: https://www.csimn.com/CSI_pages/Modbus101.html

NEVRLÝ, J. 2003. Modelování pneumatických systémů. Akademické nakladatelství CERM, s.r.o., Brno, 2003. ISBN 80-7204-300-5.

SCADA/HMI systém PROMOTIC [online]. Tavičská 845/21 703 00 Ostrava-Vítkovice: MICROSYS, spol. s r. o., c2016 [cit. 2016-06-16]. Dostupné z: <http://www.promotic.eu/cz/index.htm>.

PRZECZEK, S. 2014. Návrh a realizace elektro-pneumatických úloh. Bakalářská práce. Ostrava: VŠB – Technická univerzita Ostrava, Univerzitní studijní programy, 2014, 59 s.

Robomart: Getting Started with Arduino Ethernet #1 [online]. 2015 [cit. 2016-06-25]. Dostupné z: <https://www.robomart.com/blog/getting-started-with-arduino-ethernet>.

RONĚŠOVÁ, A. *Přehled protokolu MODBUS* [online]. 2005, 19 [cit. 2017-01-20]. Dostupné z: <http://home.zcu.cz/~ronesova/bastl/files/modbus.pdf>.

ZMIJKA, M. 2013. Řízení pneumatického pohonu pomocí signálového procesoru. Diplomová práce. VŠB - Technická univerzita Ostrava, Katedra 352, 2013, 81 s.

Přílohy

- 1) Program mikroprocesoru Arduino
- 2) Schéma zapojení úlohy jednočinný válec
- 3) Schéma zapojení úlohy dvojčinný válec
- 4) Schéma zapojení úlohy licí stroj
- 5) Schéma zapojení úlohy lis

Příloha 1 – Program mikroprocesoru Arduino

```
#include <Ethernet.h>
#include <Modbus.h>
#include <ModbusIP.h>

//Modbus Registers Offsets (0-9999)
const int SWITCH_ISTS_O_K1 = 106;
const int SWITCH_ISTS_O_K2 = 107;
const int SWITCH_ISTS_O_K3 = 108;
const int SWITCH_ISTS_O_K4 = 109;
const int SWITCH_ISTS_O_K5 = 110;
const int SWITCH_ISTS_O_K6 = 111;
const int SWITCH_ISTS_O_CR1 = 116;
const int SWITCH_ISTS_O_CR2 = 117;
const int SWITCH_ISTS_O_CR3 = 118;
const int SWITCH_ISTS_O_CR4 = 119;
const int SWITCH_ISTS_I_K1 = 100;
const int SWITCH_ISTS_I_K2 = 101;
const int SWITCH_ISTS_I_K3 = 102;
const int SWITCH_ISTS_I_K4 = 103;
const int SWITCH_ISTS_I_K5 = 104;
const int SWITCH_ISTS_I_K6 = 105;
const int SWITCH_ISTS_I_CR1 = 112;
const int SWITCH_ISTS_I_CR2 = 113;
const int SWITCH_ISTS_I_CR3 = 114;
const int SWITCH_ISTS_I_CR4 = 115;
const int SWITCH_ISTS_I_TL1 = 120;
const int SWITCH_ISTS_I_TL2 = 121;
const int SWITCH_ISTS_I_TL3 = 122;

//Used Pins
const int switchPin_O_K1 = 32;
const int switchPin_O_K2 = 34;
const int switchPin_O_K3 = 36;
const int switchPin_O_K4 = 38;
const int switchPin_O_K5 = 40;
const int switchPin_O_K6 = 42;
const int switchPin_O_CR1 = 29;
const int switchPin_O_CR2 = 31;
const int switchPin_O_CR3 = 33;
```

```
const int switchPin_O_CR4 = 35;
const int switchPin_I_K1 = 44;
const int switchPin_I_K2 = 22;
const int switchPin_I_K3 = 24;
const int switchPin_I_K4 = 26;
const int switchPin_I_K5 = 28;
const int switchPin_I_K6 = 30;
const int switchPin_I_CR1 = 37;
const int switchPin_I_CR2 = 23;
const int switchPin_I_CR3 = 25;
const int switchPin_I_CR4 = 27;
const int switchPin_I_TL1 = 45;
const int switchPin_I_TL2 = 50;
const int switchPin_I_TL3 = 49;

//ModbusIP object
ModbusIP mb;

void setup() {
    // The media access control (ethernet hardware) address for the
shield
    byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
    // The IP address for the shield
    byte ip[] = { 192, 168, 1, 120 };
    //Config Modbus IP
    mb.config(mac, ip);
    //Set Pin mode
    pinMode(switchPin_O_K1, OUTPUT);
    pinMode(switchPin_O_K2, OUTPUT);
    pinMode(switchPin_O_K3, OUTPUT);
    pinMode(switchPin_O_K4, OUTPUT);
    pinMode(switchPin_O_K5, OUTPUT);
    pinMode(switchPin_O_K6, OUTPUT);
    pinMode(switchPin_O_CR1, OUTPUT);
    pinMode(switchPin_O_CR2, OUTPUT);
    pinMode(switchPin_O_CR3, OUTPUT);
    pinMode(switchPin_O_CR4, OUTPUT);
    pinMode(switchPin_I_K1, INPUT);
    pinMode(switchPin_I_K2, INPUT);
    pinMode(switchPin_I_K3, INPUT);
    pinMode(switchPin_I_K4, INPUT);
```

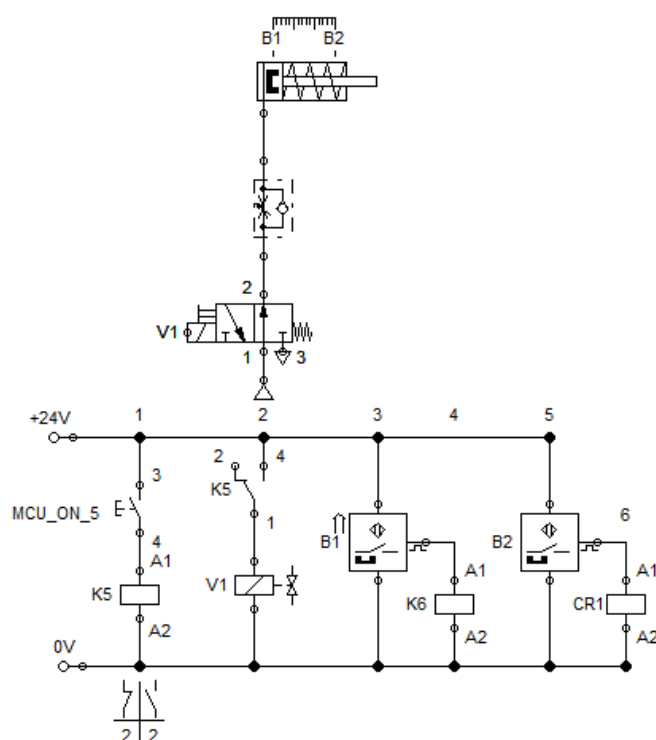
```
pinMode(switchPin_I_K5, INPUT);
pinMode(switchPin_I_K6, INPUT);
pinMode(switchPin_I_CR1, INPUT);
pinMode(switchPin_I_CR2, INPUT);
pinMode(switchPin_I_CR3, INPUT);
pinMode(switchPin_I_CR4, INPUT);
pinMode(switchPin_I_TL1, INPUT);
pinMode(switchPin_I_TL2, INPUT);
pinMode(switchPin_I_TL3, INPUT);
// Add SWITCH_ISTS register
mb.addIsts(SWITCH_ISTS_I_K1);
mb.addIsts(SWITCH_ISTS_I_K2);
mb.addIsts(SWITCH_ISTS_I_K3);
mb.addIsts(SWITCH_ISTS_I_K4);
mb.addIsts(SWITCH_ISTS_I_K5);
mb.addIsts(SWITCH_ISTS_I_K6);
mb.addIsts(SWITCH_ISTS_I_CR1);
mb.addIsts(SWITCH_ISTS_I_CR2);
mb.addIsts(SWITCH_ISTS_I_CR3);
mb.addIsts(SWITCH_ISTS_I_CR4);
mb.addIsts(SWITCH_ISTS_I_TL1);
mb.addIsts(SWITCH_ISTS_I_TL2);
mb.addIsts(SWITCH_ISTS_I_TL3);
mb.addCoil(SWITCH_ISTS_O_K1);
mb.addCoil(SWITCH_ISTS_O_K2);
mb.addCoil(SWITCH_ISTS_O_K3);
mb.addCoil(SWITCH_ISTS_O_K4);
mb.addCoil(SWITCH_ISTS_O_K5);
mb.addCoil(SWITCH_ISTS_O_K6);
mb.addCoil(SWITCH_ISTS_O_CR1);
mb.addCoil(SWITCH_ISTS_O_CR2);
mb.addCoil(SWITCH_ISTS_O_CR3);
mb.addCoil(SWITCH_ISTS_O_CR4);
}

void loop() {
    //Call once inside loop()
    mb.task();

    //Attach switchPin to SWITCH_ISTS register
    mb.Ists(SWITCH_ISTS_I_K1, digitalRead(switchPin_I_K1));
```

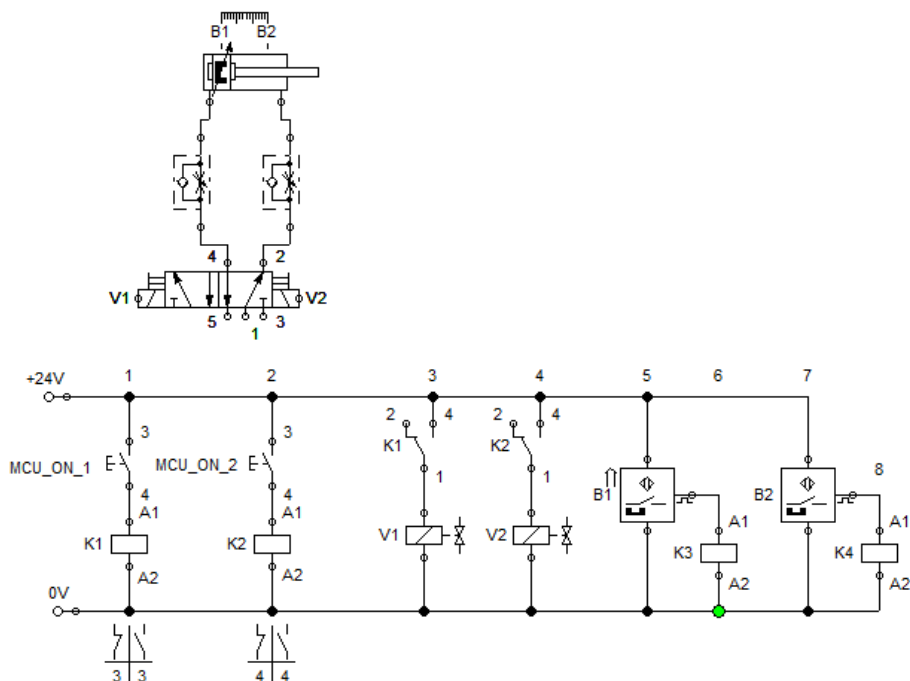
```
mb.Ists(SWITCH_ISTS_I_K2, digitalRead(switchPin_I_K2));
mb.Ists(SWITCH_ISTS_I_K3, digitalRead(switchPin_I_K3));
mb.Ists(SWITCH_ISTS_I_K4, digitalRead(switchPin_I_K4));
mb.Ists(SWITCH_ISTS_I_K5, digitalRead(switchPin_I_K5));
mb.Ists(SWITCH_ISTS_I_K6, digitalRead(switchPin_I_K6));
mb.Ists(SWITCH_ISTS_I_CR1, digitalRead(switchPin_I_CR1));
mb.Ists(SWITCH_ISTS_I_CR2, digitalRead(switchPin_I_CR2));
mb.Ists(SWITCH_ISTS_I_CR3, digitalRead(switchPin_I_CR3));
mb.Ists(SWITCH_ISTS_I_CR4, digitalRead(switchPin_I_CR4));
mb.Ists(SWITCH_ISTS_I_TL1, digitalRead(switchPin_I_TL1));
mb.Ists(SWITCH_ISTS_I_TL2, digitalRead(switchPin_I_TL2));
mb.Ists(SWITCH_ISTS_I_TL3, digitalRead(switchPin_I_TL3));
digitalWrite(switchPin_O_K1, mb.Coil(SWITCH_ISTS_O_K1));
digitalWrite(switchPin_O_K2, mb.Coil(SWITCH_ISTS_O_K2));
digitalWrite(switchPin_O_K3, mb.Coil(SWITCH_ISTS_O_K3));
digitalWrite(switchPin_O_K4, mb.Coil(SWITCH_ISTS_O_K4));
digitalWrite(switchPin_O_K5, mb.Coil(SWITCH_ISTS_O_K5));
digitalWrite(switchPin_O_K6, mb.Coil(SWITCH_ISTS_O_K6));
digitalWrite(switchPin_O_CR1, mb.Coil(SWITCH_ISTS_O_CR1));
digitalWrite(switchPin_O_CR2, mb.Coil(SWITCH_ISTS_O_CR2));
digitalWrite(switchPin_O_CR3, mb.Coil(SWITCH_ISTS_O_CR3));
digitalWrite(switchPin_O_CR4, mb.Coil(SWITCH_ISTS_O_CR4));
```

Příloha 2 – schéma zapojení úlohy jednočinný válec



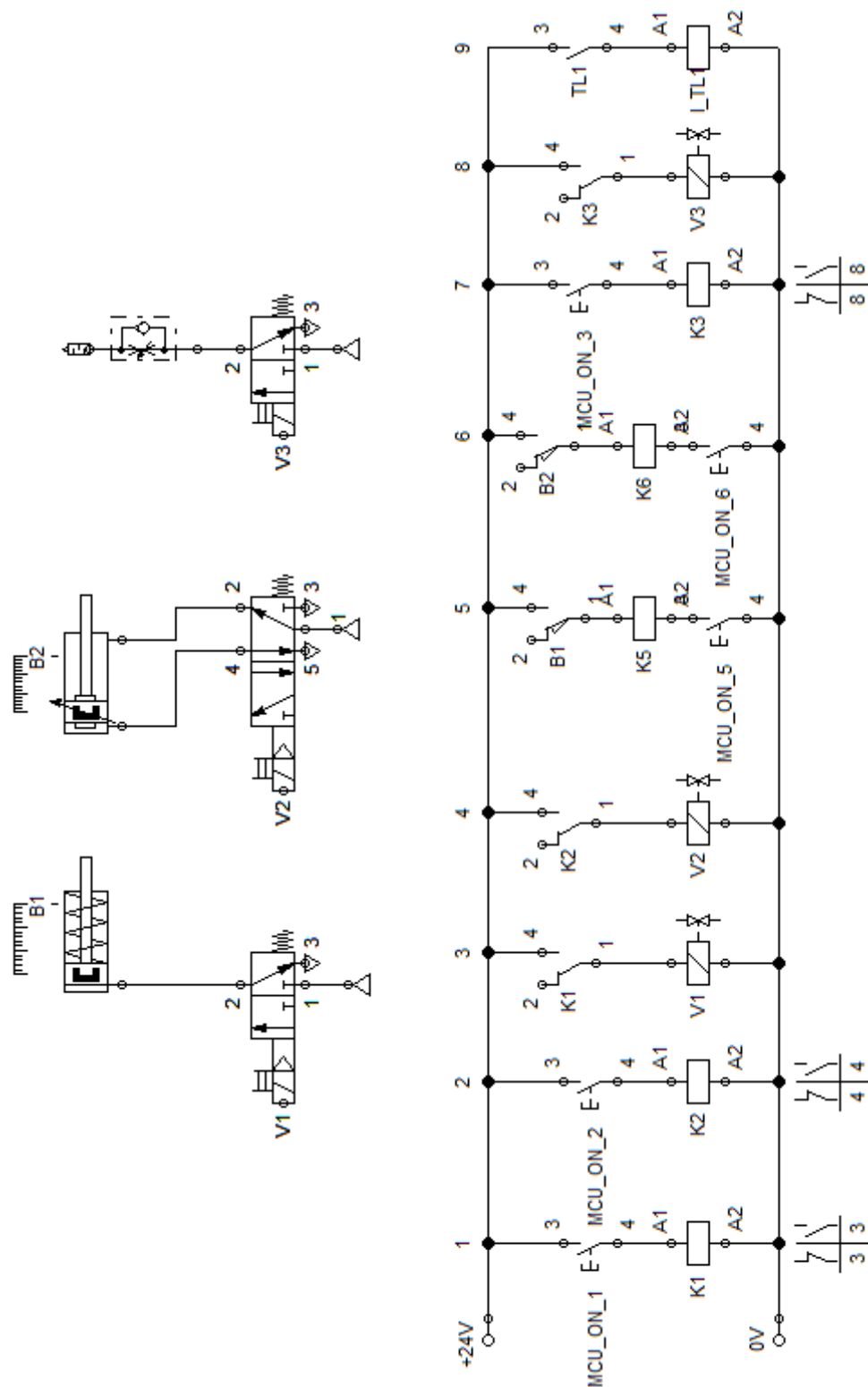
Obr. 1 Schéma zapojení - jednočinný válec

Příloha 3 – schéma zapojení úlohy dvojčinný válec



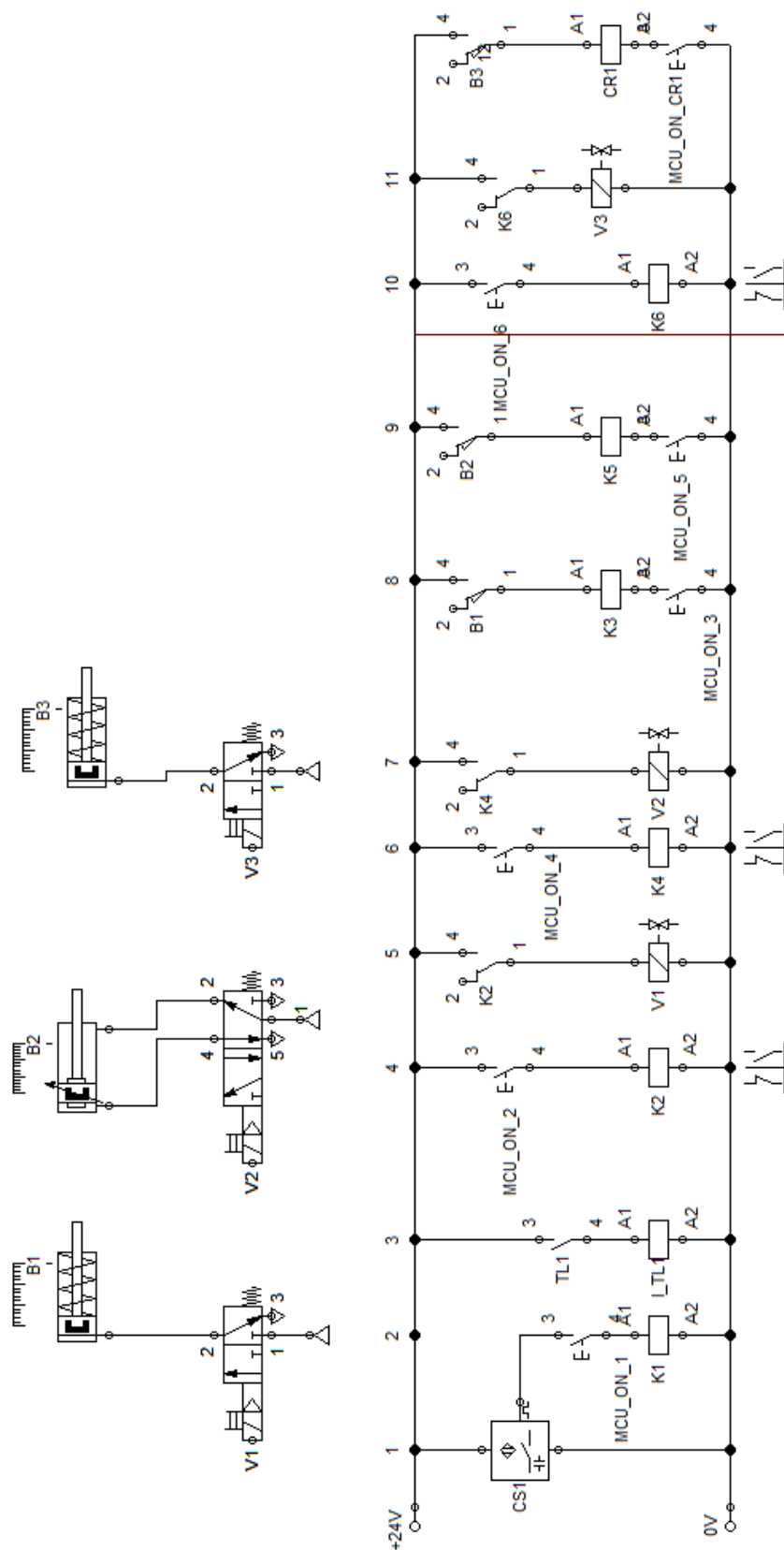
Obr. 2 Schéma zapojení - dvojčinný válec

Příloha 4 – schéma zapojení úlohy licí stroj



Obr. 3 Schéma zapojení - licí stroj

Příloha 5 – schéma zapojení úlohy lis



Obr. 4 Schéma zapojení - lis